

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES À FINALITÉ APPROFONDIE

Voyage interstellaire d'une voile solaire vers Alpha Centauri analyse de contraintes

WELCOMME, Olivier

Award date:
2020

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



UNIVERSITE DE NAMUR

FACULTÉ DES SCIENCES

Voyage interstellaire d'une voile solaire vers Alpha Centauri : analyse de
contraintes

Mémoire présenté pour l'obtention
du grade académique en master mathématiques finalité approfondie
Olivier WELCOMME
Juin 2020
Promoteur : André Füzfa

Voyage interstellaire d'une voile solaire vers Alpha Centauri : analyse de contraintes

Olivier WELCOMME

Promoteur : Pr. André Füzfa

Abstract

Ce document aborde les contraintes imposées par un voyage interstellaire, c'est-à-dire un voyage associé à un observateur accéléré voyageant à des vitesses proches de celle de la lumière. Suite à une brève introduction du sujet à l'aide du problème du voyageur de Langevin, nous introduirons le principe de la fusée à photons, en nous concentrant sur le modèle de la voile solaire. Nous travaillerons sur la cinématique de cette dernière, afin de pouvoir connaître l'évolution des différents paramètres (masse, temps propre, vitesse, ...) de la voile solaire à mesure qu'elle se déplace vers un certain système. On étendra le modèle à un modèle de voile solaire à deux étages, afin de pouvoir faire freiner la fusée pour qu'elle puisse s'arrêter sur le système cible. Finalement, à l'aide d'un formalisme Hamiltonien des équations des géodésiques obtenues depuis la métrique de Kinnersley, nous regarderons comment obtenir les trajectoires des géodésiques lumières, en prenant en compte le décalage de fréquence Doppler ainsi que les aberrations relativistes. Ce mémoire sera également accompagné d'un rapport détaillé concernant mon stage de 3 mois effectué au CURL (Cosmology, Universe and Relativity at Louvain).

This work will be centered around the interstellar traveller problem. Firstly, we will introduce the problem in special relativity, using the Twin Paradox. Secondly, we will approach the general relativity case by using a photon rocket. We will mainly stay focused on the solar sail model. We will discuss the way we can derive the radiative rocket equations to obtain an evolution of the rocketship's parameters (mass, proper time, velocity, ...). We will then approach a new model of solar sails, which is made up of two different stages. This allows us to consider a case where the solar sail is able to slow down to stop at the selected system. Finally, we will try to analyse the propagation of light towards the traveller, by obtaining the geodesic equations in the Hamiltonian formalism. This step will include the analysis of the Doppler effect and the relativistic aberration. A report of my 3 months internship at CURL (Cosmology, Universe and Relativity at Louvain) will also be available along with this master thesis.

Key words :

Interstellar travel, Photon rockets, Lightsails, Relativistic aberration

Table des matières

1	Introduction	4
2	Voyageur de Langevin	6
3	Fusée à photons	13
3.1	Introduction	13
3.2	Cinématique	14
3.2.1	Dérivation des équations de fusée à radiation	14
3.2.2	Accélération d'une fusée à photons (absorption/émission) sur un axe fixé	16
3.2.3	Solution analytique	19
3.2.4	Diffusion du laser	19
3.2.5	Effet Doppler	21
3.3	Modèle de Forward	22
3.3.1	Introduction	22
3.3.2	Comparaison de l'échelle	25
3.3.3	Diffusion du laser	27
3.3.4	Effet Doppler	28
3.3.5	Résultats finaux	30
4	Propagation de la lumière en direction de la voile solaire	34
4.1	Formalisme Hamiltonien	34

<i>TABLE DES MATIÈRES</i>	3
4.1.1 Obtention des équations géodésiques	34
4.1.2 Gestion de la masse dans le cadre des équations de géodésiques	36
4.2 Aberrations relativistes et effets Doppler	36
4.2.1 Analyse des aberrations relativistes	38
4.2.2 Analyse de l'effet Doppler	40
4.3 Applications	44
5 Conclusion	48
6 Stage	50
6.1 Déroulement du stage	50
6.2 Optimisation du code ICARUS	51
6.2.1 Introduction	51
6.2.2 Documentation du code	51
6.2.3 Optimisation sur CPU	52
6.2.4 Optimisation sur GPU	52
6.3 Développement d'un programme de visualisation de cordes cosmiques	53
6.3.1 Introduction	53
6.3.2 Lecture des données	54
6.3.3 Pré-traitement des données	55
6.3.4 Compression des données	57
6.3.5 Visualisation	57
6.3.6 Interface utilisateur	58
6.3.7 Manuel d'utilisateur	60

Chapitre 1

Introduction

Ce travail a pour but de s'interroger sur la manière par laquelle nous pourrions simuler l'envoi d'une sonde propulsée par de la lumière de la terre vers le système le plus proche : Alpha Centauri. Si on veut propulser notre sonde à l'aide de lasers, ce type de voyage implique dès lors l'utilisation de ce qu'on appelle des voiles solaires : dispositif de propulsion qui utilisera la pression de radiation émise par le laser afin de se déplacer vers la direction ciblée.

Ce type de déplacement s'accompagnant en général de vitesses comparables à celle de la lumière, il sera donc important de prendre en compte les différents effets relativistes pouvant entrer en jeu lors de tels trajets. Nous aborderons dès lors un cas d'école : le voyageur de Langevin (paradoxe des jumeaux), en suivant l'approche de E. Gourgoulhon[5], nous permettant de s'approprier des premières bases d'un voyage interstellaire à accélération constante dans le cadre de la relativité restreinte. Bien que le problème d'une fusée à accélération constante en relativité restreinte puisse être abordé sous plusieurs aspects, dont notamment le calcul des aberrations déformant la sphère céleste du voyageur (développé notamment par C. Lagoute et E. Davoust[8]), nous ne nous attarderons pas sur ces cas-là et embrayerons directement sur le modèle qui nous préoccupe majoritairement : la voyage d'une fusée à photons en relativité générale.

Pour se faire, nous baserons principalement notre analyse sur les articles de A. Füzfa[4] et J. Podolský[11], s'attaquant tous deux au problème d'un voyage interstellaire effectué par une voile solaire, à l'aide de la métrique de Kinnersley. Nous aurons pour cela besoin de connaître la manière dont se structure l'espace-temps autour du voyageur, son accélération impliquant l'utilisation d'une métrique particulière. Cela pourra être effectué à travers l'usage de la métrique de Kinnersley[7], métrique généralisant celle de Vaidya[15] (la métrique de Vaidya étant particulièrement bien développée notamment par l'article de R. W. Lindquist, R. A. Schwartz, et C. W. Misner[12]).

Ainsi, nous nous baserons sur cette métrique, avec l'appui des équations de photon rocket accessibles notamment à l'aide de W. B. Bonnor[2] et T. Damour[3], afin d'accéder à un modèle

d'équations intégrables nous permettant d'effectuer les premières intégrations, dans le but de faire évoluer la masse, la vitesse, la distance ou encore le temps inertiel en fonction du temps propre du voyageur.

Ces intégrations terminées, nous nous interrogerons sur la manière par laquelle il pourrait être possible de faire freiner le voyageur, localisé à des années-lumière de la terre. C'est également le questionnement qu'a eu R. L. Forward[14], qui a dès lors créé un prototype de voile solaire à deux (ou trois, en fonction du modèle) étages capable de faire des allers-retours vers d'autres systèmes. C'est le prototype que nous tenterons d'appliquer au modèle de voile solaire que nous aurons alors analysé. Nous comparerons les valeurs obtenues à l'aide de notre intégrateur et celles prédites par R. L. Forward, afin de voir si celles-ci semblent plausibles.

Une fois cela effectué, nous tâcherons d'appliquer davantage de contraintes au voyage, afin d'obtenir un modèle plus réaliste. Nous attaquerons alors le problème du décalage Doppler en relativité, et celui de la diffusion du laser impliquant une baisse de puissance. Le modèle de perte d'efficacité du laser sera alors analysé depuis l'article de N. Kulkarni, P. Lubin, et Q. Zhang[10], proposant un modèle prenant en compte la taille de la voile, le diamètre du laser et la longueur d'onde de ce dernier.

Nous aurons alors des résultats nouveaux, rassemblant différentes contraintes afin de pouvoir simuler le voyage d'une voile à double étage (prototype de R. L. Forward) vers Alpha Centauri.

Nous nous attaquerons ensuite à la manière dont les photons peuvent être perçus par un appareil embarqué dans le vaisseau. Dans ce cadre, nous utiliserons la solution de Kinnersley en relativité générale afin d'acquérir les expressions décrivant la trajectoire des géodésiques recherchées. Cela sera effectué en employant un formalisme Hamiltonien.

De cette approche découlera une étude plus détaillée de l'aberration relativiste perçue par l'observateur accéléré sur les signaux lumineux, cette aberration correspondant à une déviation angulaire entre les positions observées sur une sphère céleste de référence et celle de l'observateur.

S'en suivra en dernier lieu une analyse du décalage Doppler que pourront témoigner ces signaux lumineux, afin de savoir quel décalage de fréquence sera observé sur les photons, entre leur émission par des sources lumineuses (telles que des étoiles) et leur réception par l'observateur accéléré.

Chapitre 2

Voyageur de Langevin

Afin d'effectuer une première approche du problème du voyageur interstellaire, nous aborderons un modèle bien connu : le voyageur de Langevin¹, aussi appelé paradoxe des jumeaux, prenant place en relativité restreinte. Nous introduirons deux observateurs, nommés O et O' . L'observateur O sera lié à la ligne d'univers \mathcal{L} , et sera au repos sur Terre. L'observateur O' , lui, sera associé à la ligne d'univers \mathcal{L}' et correspondra à un voyageur dont la tâche sera d'effectuer une aller-retour vers un système lointain, à l'aide d'une accélération constante. Ainsi, les lignes d'univers \mathcal{L} et \mathcal{L}' seront confondues avant et après le départ du voyageur.

O étant au repos, et ne subissant par conséquent pas d'accélération, sa 4-vitesse \vec{u} sera constante. Considérons à présent le système de coordonnées affines de \mathcal{E} ($x^0 = ct, x^1 = x, x^2 = y, x^3 = z$) défini par la base orthonormale $(\vec{e}_0, \vec{e}_1, \vec{e}_2, \vec{e}_3)$ et d'origine A , où \vec{e}_0 égale la 4-vitesse de \mathcal{L} (afin de considérer le temps s'écoulant pour O comme première composante de cette base) et où A est un point commun à \mathcal{L} et \mathcal{L}' , à partir duquel l'accélération de \mathcal{L}' débute. Ainsi, la ligne \mathcal{L} étant une droite, on peut être certain que quelque soit le point M pris sur cette droite \mathcal{L} , le vecteur \vec{AM} équivaudra à $ct\vec{e}_0$, où nous rappellerons que \vec{e}_0 correspond à la 4-vitesse de O .

Ainsi, le voyageur quittera son origine en A (origine du système de coordonnées affines défini ci-dessus), et accélérera dans la même direction spatiale, que nous choisirons comme étant celle de \vec{e}_1 , jusqu'à ce qu'il entame son demi-tour, au point C_1 de la ligne d'univers \mathcal{L}' . Une fois ce point C_1 atteint, il commencera à accélérer dans la direction opposée, faisant ainsi entièrement demi-tour. Finalement, sur le retour, l'observateur O' se rapprochant de l'observateur O , nous inverserons une dernière fois la direction de l'accélération afin de retrouver une ligne d'univers \mathcal{L}' coïncidant avec la ligne \mathcal{L} , atteinte au point B .

Nous pouvons découper \mathcal{L}' en trois parties différentes, ayant chacune une forme propre à elle. Ainsi, nous considérerons les formes sur AC_1 , C_1C_2 et C_2B , la direction de la poussée étant

1. La majorité des calculs effectués ici proviennent du livre de relativité restreinte d'Éric Gourgoulhon[5]

différente pour chacune de ces parties. Ces trois parties peuvent être déterminées comme des arcs d'hyperboles, que nous pouvons définir en coordonnées affines (ct, x, y, z) de la manière suivante :

$$\text{si } t \in [0, T/4], x(t) = \frac{cT}{\alpha} \left[\sqrt{1 + \alpha^2 (t/T)^2} - 1 \right] \quad (2.1)$$

$$\text{si } t \in [T/4, 3T/4], x(t) = \frac{cT}{\alpha} \left[-\sqrt{1 + \alpha^2 (t/T - 1/2)^2} + 2\sqrt{1 + \frac{\alpha^2}{16}} - 1 \right] \quad (2.2)$$

$$\text{si } t \in [3T/4, T], x(t) = \frac{cT}{\alpha} \left[\sqrt{1 + \alpha^2 (t/T - 1)^2} - 1 \right] \quad (2.3)$$

où T est le temps propre de O écoulé entre les événements A et B , et où α est un paramètre sans dimension représentant la vitesse. Comme on le remarque sur la figure 2.1, lorsque α est nul, la ligne \mathcal{L} coïncide avec la ligne \mathcal{L}' . Au contraire, lorsque α tend vers l'infini, l'intervalle

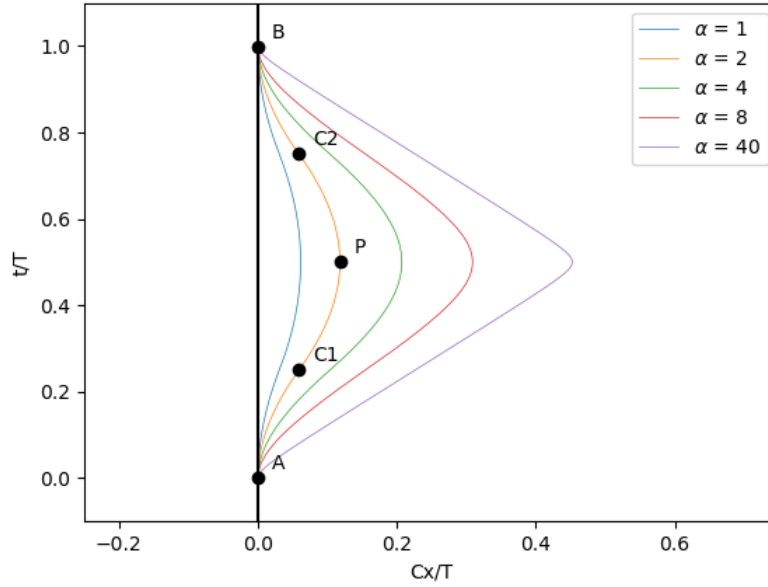


FIGURE 2.1 – En noir, la droite passant par A et B représente la ligne d'univers \mathcal{L} associée à l'observateur O' . On remarquera que pour O , la trajectoire spatiale sera celle d'une particule restant à la même position. Les 5 autres courbes en couleur représentent la forme qu'aura la ligne d'univers \mathcal{L}' associée au voyageur. Ainsi, lorsque α augmentera, on se rapprochera de plus en plus du cône de lumière, sans jamais être en mesure de l'atteindre pour autant. Ainsi, le fait de se rapprocher du cône de lumière implique un Δs (ou un intervalle espace-temps) plus faible, impliquant à son tour que le temps propre à O' s'écoulera bien plus lentement que le temps propre de O . Les points A , C_1 , C_2 et B correspondent respectivement au départ du voyageur, aux deux inversions de poussée, et à la fin du trajet de l'observateur O' . Le point P correspond donc à l'évènement mi-voyage. (Cette figure a été inspirée de la figure 2.5 du livre de relativité restreinte d'Éricourgoulhon [5], et a été réalisée à l'aide du langage Python.)

espace-temps séparant le point A du point B tendra vers 0, impliquant un voyage presque instantané pour le voyageur.

Le temps propre de O' , est un peu plus complexe à calculer que le temps propre t de O . La trajectoire spatiale circulant le long d'un axe, on sait déjà qu'un déplacement $d\vec{x}'$ sur la ligne d'univers \mathcal{L} pourra être exprimé à l'aide de la base orthonormale (\vec{e}_α) de la manière suivante : $dx'^\alpha = (cdt, dx, 0, 0)$. En effet, il sera traduit par un petit déplacement dans le temps (cdt) ainsi que sur l'axe lié à la composante \vec{e}_1 . Ce déplacement dx peut être exprimé par une différenciation des équations 2.1-2.3 citées plus haut :

$$dx = (-1)^k \frac{\alpha(t/T - k/2)}{\sqrt{1 + \alpha^2(t/T - k/2)^2}} c dt \quad (2.4)$$

Où k pourra prendre les valeurs 0, 1 et 2 en fonction de l'arc de cercle en question ($k = 0$ si $0 \leq t \leq T/4$, $k = 1$ si $T/4 \leq t \leq 3T/4$, et finalement, $k = 2$ si $3T/4 \leq t \leq T$). Ensuite, l'expression de l'évolution du temps propre t' peut être retrouvée à partir de la formule suivante :

$$dt' = \frac{1}{c} \sqrt{(dx'^0)^2 - (dx'^1)^2 - (dx'^2)^2 - (dx'^3)^2} \quad (2.5)$$

O' ne circulant que le long de l'axe x , nous pouvons assumer que dx'^2 et dx'^3 sont nuls. Or, dx'^0 et dx'^1 correspondant respectivement à ct et x , nous pouvons réduire l'expression à la suivante :

$$dt' = \frac{1}{c} \sqrt{c^2 dt^2 - dx^2} \quad (2.6)$$

En substituant dx , nous pouvons ré-écrire l'expression ainsi :

$$dt' = \frac{dt}{\sqrt{1 + \alpha^2(t/T - k/2)^2}} \quad (2.7)$$

Pour les calculs suivants, nous emploierons l'expression $\operatorname{argsinh} x$, dont la valeur vaut $\ln(x + \sqrt{x^2 + 1})$. Ce que nous désirons actuellement faire est une intégration de l'expression que nous avons trouvé pour dt' , de manière à pouvoir exprimer t' en fonction de t . En procédant à un changement de variable de la manière suivante : $\alpha(t/T - k/2) = \sinh u$, il devient aisé d'intégrer l'expression de dt' , ce qui nous donnera l'expression suivante, pour t' , exprimée en fonction du temps propre t :

$$t' = \frac{T}{\alpha} \left\{ \operatorname{argsinh} \left[\alpha \left(\frac{t}{T} - \frac{k}{2} \right) \right] + 2k \operatorname{argsinh} \left(\frac{\alpha}{4} \right) \right\} \quad (2.8)$$

Avec $k = 0$ si $0 \leq t \leq T/4$, $k = 1$ si $T/4 \leq t \leq 3T/4$ et $k = 2$ si $3T/4 \leq t \leq T$.

Notre but étant de comparer le temps propre écoulé pour le voyageur (le temps t' de l'observateur O') et le temps écoulé pour le référentiel de l'origine (le temps t de l'observateur O), nous sommes désormais en mesure de représenter graphiquement l'évolution de t' en fonction de t . Ce graphique est représenté à la figure 2.2. On remarquera une caractéristique commune à chacune de ces courbes, quelle que soit l'accélération observée : lorsque $t = T/2$, on observera systématiquement un coefficient angulaire égal à 1, signifiant un écoulement du temps identique pour les deux observateurs différents. Ce phénomène peut être expliqué par le fait

qu'à mi-trajet, le demi-tour s'effectue. Dès lors, on observe un dx nul à ce moment précis, en remplaçant t/T par $1/2$ dans l'équation 2.4. Cela implique qu'aucun déplacement de l'observateur O' n'est effectué à cet instant, par rapport à l'observateur O .

Maintenant le rapport entre l'évolution des temps ayant été analysé, nous souhaitons uniquement regarder le rapport entre les durées totales du trajet, ressenties pour chacun des deux observateurs. Ce rapport peut être facilement observé sur la figure 2.2 en regardant la valeur de t'/T au point B diminué de cette même valeur au point A . Sur base de l'équation 2.8, nous savons déjà que la valeur de $t'(B)$ peut être exprimée de la manière suivante :

$$t'(B) = \frac{4T}{\alpha} \operatorname{argsinh} \left(\frac{\alpha}{4} \right) \quad (2.9)$$

Le rapport des durées du voyage est donc facile à effectuer, $t'(B)$ étant égal à T' , et nous

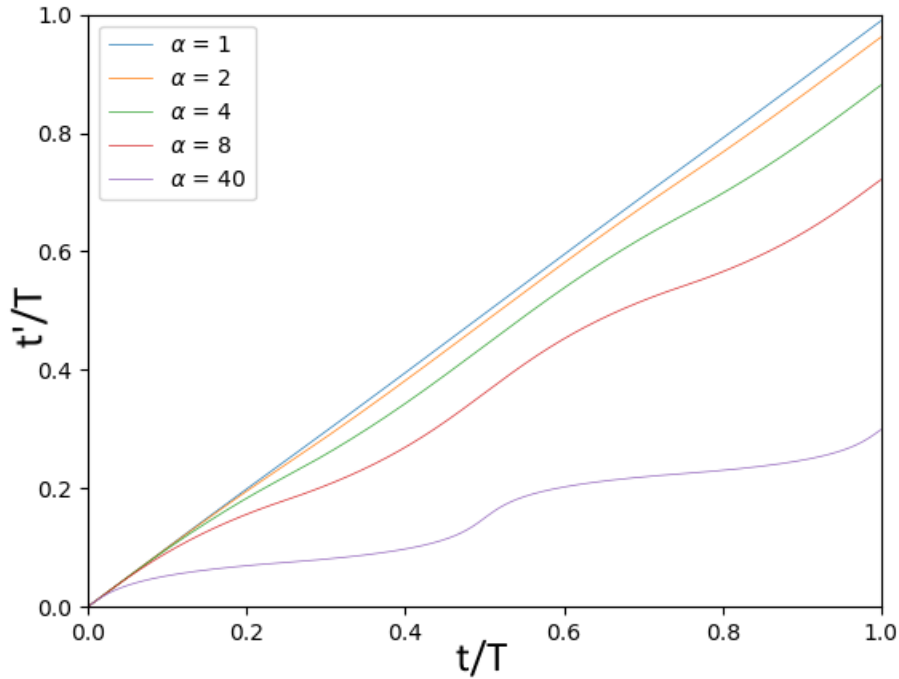


FIGURE 2.2 – Les courbes représentent l'évolution du temps propre t' de l'observateur O' en fonction du temps propre t de O . Lorsque le α sera petit, le temps propre de O' s'écoulera similairement à O . En revanche, lorsque α grandit, on remarque une dilatation du temps pour l'observateur O' . En $t/T = 1/2$, on remarquera cependant que le temps s'écoule de la même manière pour O que pour O' , car nous sommes à l'instant où le demi-tour s'effectue et le retour débute. (Cette figure a été inspirée de la figure 2.6 du livre de relativité restreinte d'Éricourgoulhon [5] et a été réalisée en Python.)

donne l'expression suivante :

$$\frac{T'}{T} = \frac{4}{\alpha} \operatorname{argsinh} \left(\frac{\alpha}{4} \right) \quad (2.10)$$

Ce rapport, pour une accélération positive ($\alpha > 0$), restera inférieur à 1, comme nous l'avons déduit de la figure 2.2. Ces rapports peuvent être représentés graphiquement (voir figure 2.3), en fonction de l'accélération. Ainsi, on est en mesure d'observer qu'effectivement, la dilatation du temps pour le voyageur sera de plus en plus importante à mesure que l'accélération α augmente. Pour une accélération nulle, nous aurons ainsi un rapport égal à 1, synonyme d'une durée identique du voyage tant pour l'observateur O que pour O' , ce qui coule de source, aucun déplacement n'ayant été effectué.

C'est dans ces observations que consiste le paradoxe des jumeaux (n'impliquant ironiquement aucune contradiction), dans le cadre de la relativité restreinte.

Nous sommes actuellement en mesure d'établir le temps écoulé et la distance parcourue par un voyageur en fonction d'un paramètre sans dimension α et du temps écoulé dans le référentiel

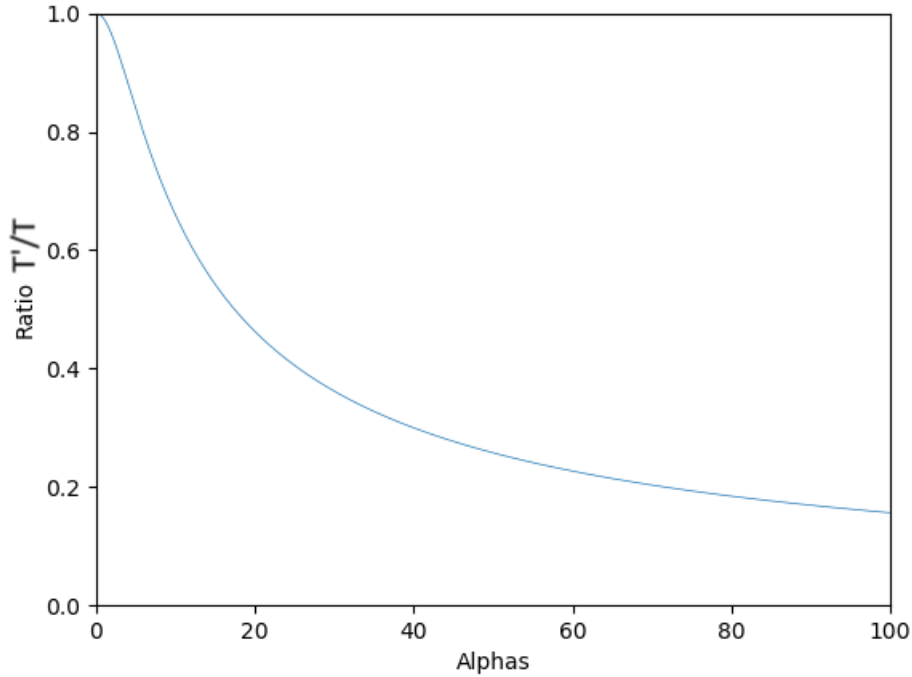


FIGURE 2.3 – Sur cette courbe est représenté le rapport entre le temps propre écoulé pour O' par rapport au temps écoulé pour l'observateur O , en fonction de l'accélération α choisie. (Cette figure a été inspirée de la figure 2.7 du livre de relativité restreinte d'Éricourgoulhon [5] et a été réalisée en Python.)

d'origine, mais nous souhaiterions pousser un peu le problème afin de pouvoir l'appliquer dans un cas réel, en faisant ainsi apparaître un paramètre possédant la dimension d'une accélération. Nous noterons ce paramètre γ , dont la valeur sera définie en fonction de α et de T de la manière suivante : $\gamma := \alpha \frac{c}{T}$. Cela nous permet d'exprimer la longueur de l'aller du voyage, ainsi que le temps écoulé pour le voyage sur base d'un paramètre à la dimension d'une accélération. Ainsi, en remplaçant le paramètre α dans l'équation 2.10, nous obtenons l'expression suivante pour le rapport entre les deux temps T' et T :

$$\frac{T'}{T} = \frac{4c}{\gamma T} \operatorname{argsinh} \left(\frac{\gamma T}{4c} \right) \quad (2.11)$$

En ce qui concerne la distance parcourue pour l'aller (qui est trivialement égale à la distance parcourue pour le retour), elle peut être obtenue à partir de l'expression de x à l'instant P , c'est-à-dire à mi-trajet, lorsque le demi-tour est effectué. Comme nous avons $k = 1$, employons dès lors l'expression 2.2 pour trouver la valeur de $x(P)$:

$$\begin{aligned} x(P) &= \frac{2cT}{\alpha} \left(\sqrt{1 + \frac{\alpha^2}{16}} - 1 \right) \\ &= \frac{\alpha}{8} \frac{cT}{\sqrt{1 + \alpha^2/16} + 1} \end{aligned} \quad (2.12)$$

en remplaçant t/T par $1/2$, étant donné que nous sommes à mi-trajet. Il ne nous reste donc plus qu'à faire un changement de variable, en remplaçant le α de manière à faire apparaître l'expression de l'accélération γ :

$$d = \frac{2c^2}{\gamma} \left[\sqrt{1 + \left(\frac{\gamma T}{4c} \right)^2} - 1 \right] \quad (2.13)$$

où $d = x(P)$ correspond donc à la distance parcourue pour l'aller ou le retour constatée lors du demi-tour en $t = T/2$. Le dernier changement que nous souhaitons faire réside dans l'expression de ces variables, non pas en fonction de T , mais de T' , afin de pouvoir exprimer la distance parcourue et le temps passé pour l'observateur O en fonction du temps écoulé pour O' . Ces expressions peuvent être obtenues en procédant à un dernier changement de variable, à l'aide d'un $T_* := \frac{4c}{\gamma} = \frac{4}{\alpha} T$. Cela nous permet donc à l'aide de l'équation 2.11 d'exprimer le temps T en fonction de T' et γ , comme souhaité :

$$T = T_* \sinh \left(\frac{T'}{T_*} \right) \quad (2.14)$$

Il ne nous reste désormais plus qu'à effectuer ce changement de variable à l'aide de la valeur de T venant d'être trouvée, et en utilisant le fait que $\cosh x = \sqrt{1 + \sinh^2 x}$, nous pouvons obtenir l'expression suivante :

$$d = \frac{cT_*}{2} \left[\cosh \left(\frac{T'}{T_*} \right) - 1 \right] \quad (2.15)$$

Nous avons donc déterminé les deux équations principales clôturant le problème du voyageur de Langevin, pour une accélération constante. Dans la littérature, il est très fréquent de choisir le paramètre γ comme étant égal à 9.81 m.s^{-2} . La raison de ce choix réside dans le fait

qu'une accélération de cet ordre-là permet de reproduire toutes les propriétés de la pesanteur terrestre.

Afin d'illustrer le problème du voyageur de Langevin à l'aide de quelques exemples, nous choisirons donc de prendre une accélération γ de valeur égale à 9.81 m.s^{-2} . Nous pouvons dès lors, à partir des équations 2.14 et 2.15, estimer le temps passé sur Terre et la distance maximale atteinte à partir du temps T' passé sur le vaisseau. Par exemple, pour ce γ , 2 ans pour le vaisseau signifiera 2.09 années passées sur Terre et un aller-retour doté d'une portée de 0.26 années-lumière. Au plus le temps passé sur le vaisseau sera important, au plus la vitesse du vaisseau augmentera, impliquant une dilatation du temps de plus en plus importante. Ainsi, en augmentant la valeur de T' , le rapport T'/T diminuera également à son tour. Par exemple, là où en 2 ans nous avons un rapport encore assez proche de 1, nous pouvons observer qu'en 16 ans sur le vaisseau, 120 années se seront écoulées sur Terre, pour une distance de 58 années-lumière, impliquant un rapport bien plus petit.

Chapitre 3

Fusée à photons

3.1 Introduction

Le sujet de ce chapitre est centré sur le principe de fusée à photons. L'idée se cachant derrière ce principe est d'octroyer une poussée à une fusée via l'émission ou l'absorption de photon (permettant dès lors une pression radiative). Dans le cadre de ce mémoire, nous serons axés sur le cas de la voile solaire.

Dans notre analyse, nous allons dans un premier temps étudier le cas simple où nous considérons une voile solaire propulsée par un laser, idée introduite en 1966 par G.Marx [9], reprise plus tard, notamment par J.L.Redding [13], J.F.L.Simmons et C.R.McInnes [6]. Nous prendrons soin, dans ce cas, d'omettre une partie des contraintes, telles que l'effet Doppler ou une diffusion plus complexe du laser, lesquelles seront étudiées par après. Une analyse d'erreur avec le modèle analytique sera effectuée afin d'avoir un indice sur la fiabilité des résultats.

Les analyses en question porteront sur l'application de la relativité générale aux fusées à photons, utilisant se faisant une solution pour l'espace-temps autour de la fusée. Cette solution est issue métrique de Kinnersley[7]. Cette dernière métrique généralise la métrique de Vaidya [15][12], métrique qui peut être obtenue depuis celle de Schwarzschild (qui décrit la géométrie de l'espace-temps déformée par le champ gravitationnel à l'extérieur d'un corps isolé) exprimée à l'aide des coordonnées d'Eddington–Finkelstein. Ce changement de coordonnées dans la métrique de Schwarzschild permet de se débarrasser de la singularité en $r = 2M$, en introduisant par la même occasion une coordonnée entrante (ou sortante) nulle " u ", qui sera reprise plus tard dans notre analyse. Nous étudierons ainsi les fluctuations de masse et de vitesse au cours du trajet, l'évolution du temps propre et le comportement des géodésiques lumières perçues par l'observateur embarqué. Cette première partie suivra majoritairement la structure adoptée dans l'article d'A. Füzfa, dans le cadre de l'intégration d'un modèle de voile solaire [4].

Dans un second temps, nous introduirons le cas proposé par Forward [14], basé sur une

voile solaire à deux étages. Nous aborderons d'abord ce cas de manière généralisée, sans dimension, comme nous l'avons fait avec le cas simple. Ensuite, nous utiliserons les paramètres utilisés dans l'article de Forward afin de pouvoir comparer les échelles de distance entre nos résultats et ceux du document original. Finalement, nous étudierons l'application des contraintes mentionnées plus haut (effet Doppler et diffusion du laser[10]) afin d'obtenir les résultats clôturant cette section sur la cinématique et le modèle de Forward.

3.2 Cinématique

3.2.1 Dérivation des équations de fusée à radiation

Dans le cadre de l'analyse de la cinématique de la fusée à photons, nous introduirons deux observateurs distincts, avec associé à chacun un système de coordonnées. Nous aurons \mathcal{O} , observateur inertiel situé très loin de la fusée de manière à ne pas subir les effets gravitationnels de la fusée, auquel nous associerons un champ inertiel lié aux coordonnées cartésiennes $(X^\mu)_{\mu=0,\dots,3} = (cT, X, Y, Z)$ où c est la vitesse de la lumière. Le second observateur \mathcal{O}' , correspond à l'observateur embarqué dans la fusée. Nous lui attribuerons les coordonnées sphériques co-mobiles suivantes : $(x^\nu)_{\nu=0,\dots,3} = (c\tau, r, \theta, \varphi)$ où τ correspond au temps propre du voyageur.

La trajectoire empruntée par la fusée pour atteindre Alpha Centauri sera considérée comme une ligne droite, le long de l'axe Z pour l'observateur \mathcal{O} . La ligne d'univers du voyageur \mathcal{L} pourra être exprimée en fonction des deux systèmes de coordonnées :

$$\mathcal{L} \equiv (cT(\tau), 0, 0, Z(\tau)) \quad (3.1)$$

dans les coordonnées de \mathcal{O} avec τ qui correspondra ici au temps propre du voyageur accéléré, ou

$$\mathcal{L} \equiv (c\tau, r = 0) \quad (3.2)$$

pour l'observateur \mathcal{O}' , son système de coordonnées étant co-mobiles. Le rayon étant nul, les valeurs de θ et φ n'auront aucune influence sur \mathcal{L} . A cette ligne d'univers \mathcal{L} nous pouvons associer un champ de vecteurs tangents, que l'on notera $\lambda^\mu = \frac{dX^\mu}{cd\tau}$ pour \mathcal{O} , et $\lambda'^\mu = \frac{dx^\mu}{cd\tau}$ pour \mathcal{O}' .

Nous pouvons utiliser les coordonnées co-mobiles (cu, r, θ, φ) afin de décrire la géométrie de l'espace-temps perçue par le voyageur \mathcal{O}' à l'aide de la métrique suivante[11][7], l'accélération de la fusée pouvant être perçue comme un champ gravitationnel local par le voyageur :

$$\begin{aligned}
ds^2 = & c^2 \left(1 - 2\frac{M}{r} - 2\alpha r \cos(\theta) - \alpha^2 r^2 \sin^2(\theta) \right) du^2 \\
& \pm 2cdudr \mp 2\alpha r^2 \sin(\theta) cdud\theta \\
& - r^2(d\theta^2 + \sin^2(\theta)d\varphi^2)
\end{aligned} \tag{3.3}$$

où le signe sera établi en fonction que la métrique soit sortante ou entrante, impactant par conséquent l'évolution de la coordonnée u associée au temps, dans les coordonnées co-mobiles (cu, r, θ, φ) . On peut remarquer que le cas où $r = 0$ impliquera une singularité dans la métrique [2]. On remarquera également que lorsque $M \rightarrow 0$, ou lorsque $r \rightarrow \infty$, la coordonnée nulle u prendra la valeur de du temps propre du voyageur accéléré τ . Nous verrons dans le chapitre suivant celui-ci que le paramètre de masse pourra être négligé, dans la mesure où la luminosité employée pour propulser la fusée demeure relativement faible ($\frac{GL}{c^5} \ll 1$). L'espace-temps autour de la fusée sera défini suivant deux fonctions : la fonction liée à la masse inertielle m , et la fonction de quadri-accélération[5], respectivement $M(u)$ et $\alpha(u)$. La fonction $M(u)$ peut être exprimée de la manière suivante :

$$M(u) = 2Gm(u)/c^2 \tag{3.4}$$

où G est la constante de Newton.

Pour la fonction de 4-accélération, nous considérons dans un premier temps $\dot{\lambda}^\mu = d\lambda^\mu/(cd\tau)$ comme la quadri-accélération de la ligne d'univers \mathcal{L} . Nous avons que le vecteur unitaire tangent λ^μ est de type temps, c'est-à-dire que la norme de ce quadri-vecteur $\eta_{\mu\nu}\lambda^\mu\lambda^\nu = 1$, la signature de la métrique étant $(+, -, -, -)$. Dès lors, on a que $\dot{\lambda}^\mu$ est un vecteur de type temps (car $\dot{\lambda}^\mu\lambda_\mu = 0$). La fonction de 4-accélération sera alors donnée par la formule suivante :

$$\alpha^2 = -\dot{\lambda}^\mu\dot{\lambda}_\mu = -\dot{\lambda}'^\mu\dot{\lambda}'_\mu \geq 0 \tag{3.5}$$

Une transition entre les différents systèmes de coordonnées (de \mathcal{O}' aux coordonnées inertielles de \mathcal{O}) peut être effectuée à travers les opérations suivantes[2] :

$$\begin{aligned}
T &= T_{\mathcal{L}}(\tau) + r[\pm \cosh(\psi) + \cos(\theta) \sinh(\psi)] \\
X &= r \sin(\theta) \cos(\varphi) \\
Y &= r \sin(\theta) \sin(\varphi) \\
Z &= Z_{\mathcal{L}} + r[\pm \sinh(\psi) + \cos(\theta) \cosh(\psi)]
\end{aligned} \tag{3.6}$$

où $T_{\mathcal{L}}(\tau)$ et $Z_{\mathcal{L}}$ correspondent à la ligne d'univers \mathcal{L} dans les coordonnées inertielles, et où $\psi(\tau)$ est associé à la rapidité. De plus, le choix du signe sera établi en fonction du fait que l'on aborde la métrique sortante (+) ou entrante (-). Ces transformations ne sont valides que pour une masse négligeable, ou à une distance infinie de la fusée[2].

Il peut également être trouvé, en considérant un $m = 0$, le comportement des vecteurs tangents et de la quadri-accélération en effectuant des transformations sur la métrique 3.3 :

$$\lambda^T = \frac{dT_{\mathcal{L}}}{d\tau} = \cosh(\psi(\tau)) \quad (3.7)$$

$$\lambda^Z = \frac{dZ_{\mathcal{L}}}{d\tau} = c \sinh(\psi(\tau)) \quad (3.8)$$

$$\alpha(\tau) = -\dot{\psi} \quad (3.9)$$

3.2.2 Accélération d'une fusée à photons (absorption/émission) sur un axe fixé

Dans le cadre de notre simulation, le voyageur ne se déplace que sur un seul axe. Dès lors, les vecteurs tangents λ^X et λ^Y seront tous deux nuls. Les équations des fusées relativistes à radiation dont nous partirons sont les suivantes[3] :

$$\dot{m}c^2 = \lambda_\mu f^\mu \quad (3.10)$$

$$mc^2 \dot{\lambda}^\mu = f^\mu - \lambda_\beta f^\beta \lambda^\mu \quad (3.11)$$

où f^μ correspond à la quadri-force[5] (f^T correspond à la puissance entrant ou sortant de la fusée, changeant de signe en fonction du cas en question).

A partir des équations 3.7 et 3.8, nous pouvons transformer la formule citée ci-dessus afin d'obtenir les équations suivantes :

$$\dot{m}c^2 = \cosh(\psi) f^T - \sinh(\psi) f^Z \quad (3.12)$$

$$mc^2 \dot{\psi} = -\sinh(\psi) f^T + \cosh(\psi) f^Z \quad (3.13)$$

La 3-vitesse peut facilement être obtenue à l'aide du rapport entre les deux vecteurs tangents λ^Z et λ^T , nous permettant à partir de 3.7 et 3.8 d'obtenir la définition suivante :

$$V^Z = \frac{dZ}{dT} = c \frac{\lambda^Z}{\lambda^T} = \frac{c \sinh(\psi(\tau))}{\cosh(\psi(\tau))} = c \tanh(\psi) \quad (3.14)$$

Il s'agit donc de la 3-vitesse par rapport au référentiel inertiel de temps propre T . Il suffira de dériver cette dernière formule afin d'obtenir la 3-accélération, toujours vis-à-vis de l'observateur inertiel \mathcal{O} , qui sera exprimée de la manière suivante :

$$a = \frac{dV^Z}{dT} = \frac{c \dot{\psi}}{\cosh^3(\psi)} \quad (3.15)$$

Maintenant la dynamique basique de la fusée établie, nous pouvons étendre sa formulation en une autre dénuée de dimension, en faisant usage d'une échelle caractéristique liée au temps propre que l'on notera $\tau_c = m_0 c^2 / |P|$, où m_0 est la masse initiale inertielle de la fusée, et où P est l'échelle de puissance entrant ou sortant de la fusée, le signe de P étant à nouveau déterminé par le fait que la puissance sorte (-) ou entre (+). Nous introduisons le temps sans dimension $s = \tau / \tau_c$, ce qui nous permet de réécrire le modèle ainsi :

$$\mathcal{M}' = \mathcal{P}(\tau) [\cosh(\psi) - \sinh(\psi) \mathcal{T}(\tau)] \quad (3.16)$$

$$\mathcal{M}\psi' = \mathcal{P}(\tau) [-\sinh(\psi) + \cosh(\psi) \mathcal{T}(\tau)] \quad (3.17)$$

où $\mathcal{M} = m/m_0$, $\mathcal{P}(\tau) = f^T/|P|$ et $\mathcal{T} = f^Z/f^T$. \mathcal{P} et \mathcal{T} correspondent respectivement à des paramètres sans dimension liés à la puissance entrante/sortante et à la poussée appliquée sur la fusée. Le signe de \mathcal{P} , comme mentionné plus haut pour P ou f^T , dépendra du fait que la fusée soit à absorption ou émission (le modèle d'absorption étant celui associé à la voile solaire).

Un résultat essentiel à mentionner est le fait que $f^\mu P_\mu = 0$. Pour rappel, f^μ correspond ici à la quadri-force associée à la fusée, et P_μ est liée à la quadri-impulsion du laser. Ce dernier résultat peut être obtenu à partir de plusieurs facteurs. Dans un premier temps, on a que la quadri-impulsion totale peut être exprimée de telle manière que $\mathcal{P}^\mu = p^\mu + P^\mu$. De plus, on a la quadri-force totale qui pourra être exprimée en dérivant la quadri-impulsion de manière à ce que $\mathcal{F}^\mu = \frac{d\mathcal{P}^\mu}{d\tau} = 0 \implies f_{voile}^\mu = -F_{laser}^\mu$, impliquant que \mathcal{P}^μ est une constante et donc que $\mathcal{P}^\mu \mathcal{P}_\mu$ est également constant. Cela signifie donc que $\frac{d(\mathcal{P}^\mu \mathcal{P}_\mu)}{d\tau}$ est nul. Finalement, la masse du photon étant nulle, on aura pour le laser que $P^\mu P_\mu = 0$.

Dans un modèle réaliste, pour la voile solaire, il faudra considérer le fait que la puissance ne sera pas constante, et qu'elle décroîtra à mesure que la fusée s'éloigne de la source de lumière. Dans le cadre de nos premiers résultats, nous considérerons dès lors que l'intensité du rayon lumineux décroîtra de la manière suivante : $\mathcal{P}(\tau) = Z^{-2}(\tau)$, dans la mesure où la source du rayon lumineux ne change pas d'intensité. Dans le cadre de notre modèle sans dimension, nous utiliserons dans un premier temps une première version très simplifiée de la décroissance de puissance du laser. Cette version sera très vite laissée de côté pour une utilisation d'une version plus réaliste dans un second temps, en adéquation avec l'article de Lubin et Kulkarni [10]. Nous aurons donc, dans notre première version, le comportement de \mathcal{P} s'exprimant ainsi : $\mathcal{P}(\tau) = (1 + Z(\tau)/Z_c)^{-2}$ où $Z_c = c\tau_c$. Le second modèle de décroissance de puissance sera détaillé plus loin, afin de faire évoluer le modèle de manière plus réaliste, prenant en compte la taille de la voile.

Concernant les fusées à émission, on peut considérer également une décroissance de la puissance liée à la baisse de chaleur d'un corps radioactif, baisse que nous caractériserons ainsi : $\mathcal{P}(\tau) = \exp(-s/S)$, où S correspond à l'échelle à laquelle la baisse de chaleur est observée (à l'aide de l'échelle caractéristique τ_c introduite plus haut).

Finalement, un dernier paramètre que nous souhaiterons aborder avant d'effectuer les premières intégrations est celui lié au facteur d'absorption des voiles solaires. Effectivement,

selon que ces dernières soient noires ou blanches, la poussée subie différera. Nous définirons alors la fonction de poussée de la manière suivante : $\mathcal{T} = \pm(1 + \varepsilon)$, où le signe correspondra à la direction de l'accélération. Un $\varepsilon = 0$ exprimera une voile parfaitement noire, et $\varepsilon = 1$ caractérisera une voile blanche. Par soucis de compatibilité avec la contrainte $f^\mu P_\mu = 0$, nous restreindrons nos modèles à l'utilisation d'une voile blanche, nous permettant par conséquent une réflexion parfaite des photons par la voile.

Nous pouvons désormais effectuer les premières intégrations. Pour celles-ci, nous aborderons dans un premier temps le modèle sans dimension. Dans les figures 3.1 et 3.2, nous considérons dès lors une masse initiale $M_0 = 1$ et une 3-vitesse initiale nulle. La puissance décroissant avec le temps, nous pouvons observer le fait que la vitesse et la masse inertielle vont subir de moins en moins de changement à mesure que le temps passe, l'impact de la puissance s'amoindrissant, la fusée atteignant une portion significative de la vitesse de la lumière.

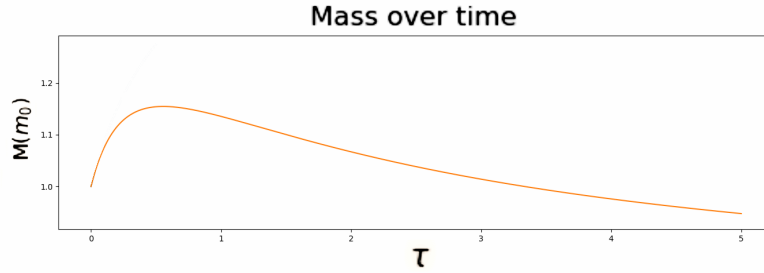


FIGURE 3.1 – Évolution du facteur sans dimension de masse inertielle en fonction du temps propre du voyageur.

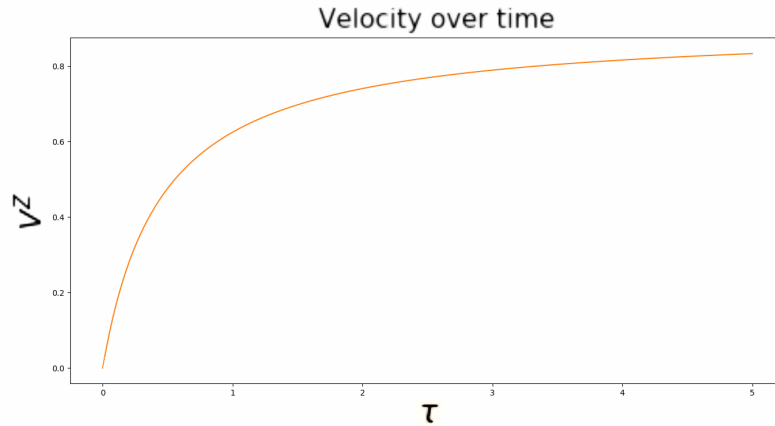


FIGURE 3.2 – Évolution de la vitesse de la fusée pour l'observateur inertiel en fonction du temps propre du voyageur.

3.2.3 Solution analytique

Dans l'optique de pouvoir vérifier de manière aisée l'intégration pour un cas trivial, nous pouvons introduire la solution analytique du système, en considérant une puissance constante avec $\mathcal{P}(\tau) = \pm 1$ et $\mathcal{T}(\tau) = \pm 1$, en considérant par conséquent une puissance directement proportionnelle à la poussée. Cette solution analytique sera alors exprimée ainsi :

$$M = (1 + 2\mathcal{P}e^{-\mathcal{P}\mathcal{T}\psi_0}(s - s_0))^{1/2} \quad (3.18)$$

$$\psi = \psi_0 + \mathcal{P}\mathcal{T} \log \mathcal{M} \quad (3.19)$$

où s_0 et ψ_0 sont respectivement le temps initial, et la rapidité, sans dimension. Cette solution, comme mentionnée précédemment, correspond à celle associée à une puissance constante. Cela implique que les effets provoquant une fluctuation de la puissance de la fusée, tels que l'effet Doppler, invalideront cette solution. Dès lors, nous ne considérerons ici cette solution que pour le cas de la fusée à émission, où l'effet Doppler n'intervient donc pas, contrairement au cas de la voile solaire. Le but principal de cette solution étant de vérifier la rigueur de notre intégration numérique, la limiter ainsi aux fusées à émission ne devrait pas altérer son usage de manière trop inconveniente.

Cette solution nous permet d'établir un graphe d'erreur à l'échelle logarithmique, visible à la figure 3.3, pour respectivement la masse et la rapidité.

3.2.4 Diffusion du laser

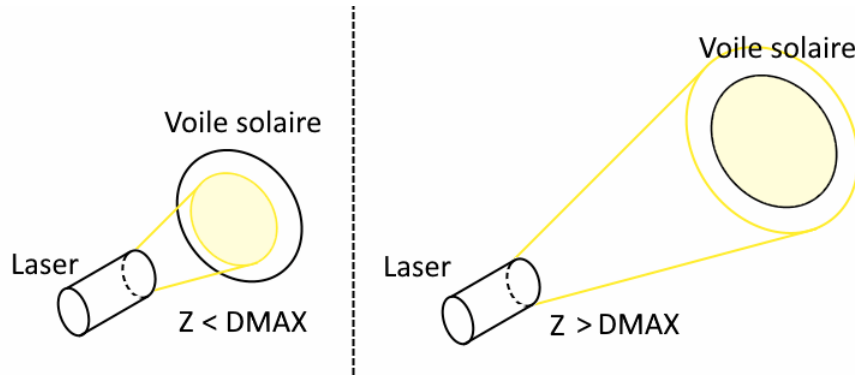


FIGURE 3.4 – Figure représentant la décroissance de puissance perçue par le vaisseau, dès que DMAX est atteint (le laser ne devient plus circonscrit dans la voile), la baisse de puissance est appliquée[10].

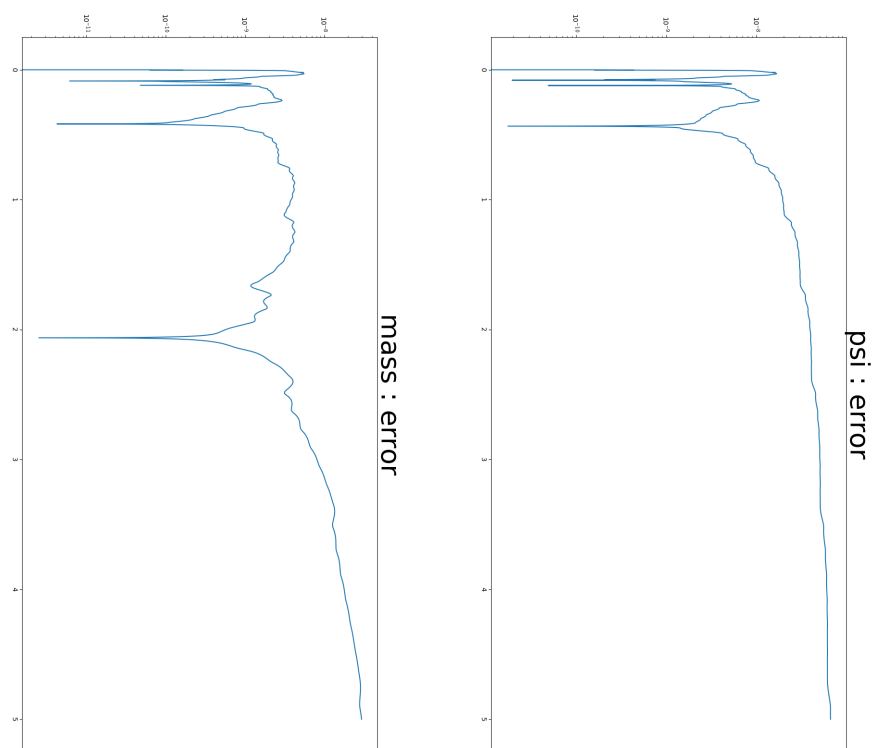


FIGURE 3.3 – Graphes d'erreurs, comparant l'intégration à la solution analytique à l'aide d'un graphe en semi-log10. Les graphes d'erreurs correspondent respectivement au facteur de masse inertielle et à la rapidité ψ .

Comme proposé dans l'article de Lubin[10], un modèle alternatif peut être utilisé vis-à-vis de la décroissance de la puissance du laser. L'idée serait non plus d'avoir une puissance qui décroît immédiatement en fonction de la distance, mais d'avoir une puissance constante jusqu'à ce que le faisceau lumineux dépasse la taille de la voile de part sa divergence. Une fois cette taille dépassée, la puissance commencerait alors seulement à décroître, comme le montre la figure 3.4.

On aura donc une distance limite, que l'on notera D_{max} , à partir de laquelle la puissance décroîtra en suivant un modèle similaire à celui vu plus haut (inverse de la distance parcourue). Cette distance repose principalement sur 3 constantes : la longueur d'onde initiale du laser, le diamètre de celui-ci et la taille de la voile. La divergence du faisceau laser sera exprimée ainsi :

$$\theta = \lambda/d \quad (3.20)$$

où θ est la divergence, λ la longueur d'onde et d le diamètre du laser. Ensuite, la distance maximale peut être trouvée en regardant à partir de quelle distance le laser sera circonscrit dans la voile :

$$D_{max} = L_{voile}/\theta \quad (3.21)$$

Dans l'article de Lubin, on considère une longueur d'onde de 1064 nm et un diamètre du laser de 10km. En considérant une longueur de voile de 10m, on aurait un angle de divergence $\theta = 10^{-10}rad$ ce qui nous mène à une distance maximale de $10^8 km$. Par souci de simplicité, les valeurs liées à la longueur d'onde et au diamètre du laser sont celles que nous utiliserons dans notre intégration. En considérant les paramètres de l'article de Forward [14] (voile de 1000km), on aurait pour le cas qui nous intéresse une distance maximale de $1e13 km$, avant que la puissance ne commence à décroître. Une fois cette distance atteinte, on exprimera le facteur réduisant la puissance de la manière suivante :

$$\left(\frac{D_{max}}{Z(\tau)}\right)^2 \quad (3.22)$$

3.2.5 Effet Doppler

Par souci de réalisme, nous rajouterons également un facteur supplémentaire, pouvant avoir un impact majeur sur le comportement de la voile solaire : le décalage Doppler. Analysons dès lors l'effet produit par le décalage en fréquence de la lumière, lié à la vitesse du vaisseau. Le facteur longitudinal de Doppler relativiste sera généralement exprimé par le rapport suivant :

$$\sqrt{\frac{1+\beta}{1-\beta}} \quad (3.23)$$

Cependant, on sait que $V^Z = c \tanh(\psi)$, et donc que $\beta = \tanh(\psi)$, ce qui permet simplement de ré-exprimer le facteur de Doppler ainsi :

$$\sqrt{\frac{1 + \tanh(\psi)}{1 - \tanh(\psi)}} \quad (3.24)$$

C'est le facteur qui sera alors utilisé dans l'expression de la fonction de puissance f^Z de notre fusée. En conclusion, si on reprend le décalage de fréquence et la diffusion du laser sous le modèle de Lubin, on a modifié la fonction de puissance sous cette forme finale :

$$f^Z = (1 + \epsilon) * \sqrt{\frac{1 + \tanh(\psi)}{1 - \tanh(\psi)}} * P_{source} * 1 \quad (3.25)$$

si $Z(\tau) \leq D_{max}$, ou sinon :

$$f^Z = (1 + \epsilon) * \sqrt{\frac{1 + \tanh(\psi)}{1 - \tanh(\psi)}} * P_{source} * \left(\frac{D_{max}}{V^Z}\right)^2 \quad (3.26)$$

3.3 Modèle de Forward

3.3.1 Introduction

Nous allons désormais introduire le modèle de Forward[14], modèle basé sur celui de la voile solaire simple, le développant sur un modèle à double voile, centré sur une fusée à deux étages.

Le questionnement qui a mené Forward à envisager ce modèle était de se demander par quelle manière il peut être possible de faire freiner une voile solaire déjà loin de sa source lumineuse, sans avoir à compter sur une autre source de puissance. Le principe qu'il a alors avancé était celui d'une fusée à deux étages : un premier étage composé d'une petite voile circulaire et d'un payload en son centre, et un second étage composé d'une grande voile annulaire, circulant sur le pourtour de la petite voile. Dans un premier temps, les première et seconde voiles ne forment qu'une voile, voyageant jusqu'à être à proximité de la destination. Une fois suffisamment proche de la destination, la séparation des deux étages prend place, et les rayons lumineux viennent se refléter sur la grande voile (le second étage) pour venir se heurter de l'autre côté du premier étage (contenant le payload), et faisant ainsi accélérer la petite voile dans la direction opposée à sa destination, faisant ainsi baisser la vitesse du premier étage. Une représentation du fonctionnement de ce vaisseau est visible sur la figure 3.5

Bien utilisé, ce modèle permet donc d'envoyer une voile solaire vers Alpha Centauri, dont le payload pourrait être livré en orbite dans le système de destination.

Dans le cadre de notre analyse, nous allons tenter d'appliquer dans un premier temps le modèle vu sur la voile solaire simple, et de comparer les échelles avec la solution proposée par Forward, avant de complexifier le modèle avec le rajout de l'effet Doppler et d'une diffusion de laser plus complexe. Notons τ_i le temps où la séparation des deux anneaux sera effectuée, pour le voyageur.

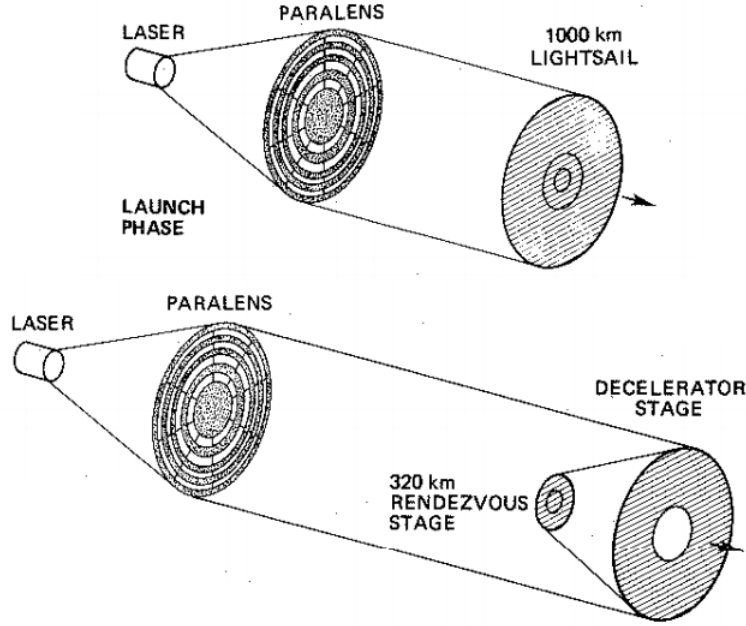


FIGURE 3.5 – Figure représentant le vaisseau et ses différentes voiles. Cette figure est tirée de l'article de Forward [14]

La masse initiale du vaisseau sera de $m_0 = m_{i0} + m_{e0}$, où m_{i0} et m_{e0} sont respectivement la masse initiale de la voile intérieure et extérieure. Le vaisseau se comportera comme une voile solaire standard jusqu'à atteindre τ_i , où chacun des deux étages se comportera comme un vaisseau indépendant. Le rapport des masses entre les anneaux étant conservé à travers les variations de masse subies jusque τ_i , on peut en déduire qu'à l'instant τ_i , les masses de l'anneau extérieur et intérieur seront exprimées ainsi :

$$m_{e\tau_i} = m(\tau_i) \cdot \left(\frac{m_{e0}}{m(\tau_0)} \right) \quad (3.27)$$

$$m_{i\tau_i} = m(\tau_i) \cdot \left(\frac{m_{i0}}{m(\tau_0)} \right) \quad (3.28)$$

Le modèle de \mathcal{P} vu plus haut sera toujours valable pour l'anneau extérieur, n'ayant de différence par rapport au modèle classique que dans la perte d'une fraction de sa masse à un

instant τ_i . Cependant, pour l'anneau intérieur, la distance parcourue par la lumière n'évoluera pas exactement de la même manière, la lumière devant rebondir sur la voile extérieure avant de venir impacter celle intérieure :

$$\mathcal{P}_i(\tau) = \frac{1}{(Z_e(\tau) + (Z_e(\tau) - Z_i(\tau))^2)} \quad (3.29)$$

En reprenant le modèle sans dimension proposé précédemment, nous pouvons à présent intégrer cette première version de la solution. Pour reprendre le contexte proposé par Forward, nous considérerons ici que la voile extérieure compose 90% de la masse totale du vaisseau. Comme on peut l'observer sur les figures 3.6 et 3.7, les deux voiles poursuivent l'exacte même trajectoire jusqu'à leur séparation, où la voile intérieure va petit à petit freiner jusqu'à avoir une vitesse nulle pendant que la voile extérieure poursuit son parcours comme une voile standard.

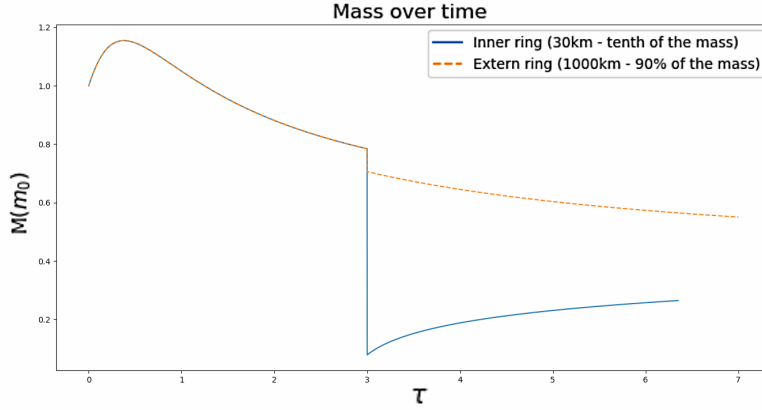


FIGURE 3.6 – Évolution de la masse des voiles solaires pour l'observateur inertiel en fonction du temps propre du voyageur. La courbe orange sera associée à la voile intérieure, composant 10% de la masse totale du vaisseau, et la courbe bleue sera celle de la voile extérieure.

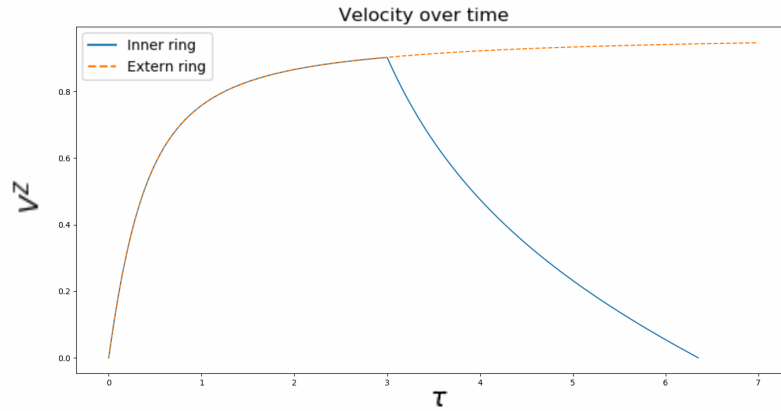


FIGURE 3.7 – Évolution de la 3-vitesse des voiles solaires pour l’observateur inertiel en fonction du temps propre du voyageur. La courbe orange sera associée à la voile intérieure, composant 10% de la masse totale du vaisseau, et la courbe bleue sera celle de la voile extérieure.

3.3.2 Comparaison de l’échelle

Concrétisons désormais davantage notre simulation, en utilisant le modèle avec des valeurs plausibles. Par souci de simplicité et de vérification, nous emploierons les mêmes valeurs que celles proposées dans la solution de Forward. Ainsi, nous considérons une fusée dont la masse initiale est de 785 tonnes. Le papier de Forward souhaite octroyer à la fusée une accélération constante de $0.05m/s^2$, la source gagnant en intensité au fur et à mesure des années, passant de $7.2TW$ au début du voyage à $26TW$ lors de la séparation. Cependant, nous limiterons notre contexte à une source d’énergie constante fixée à $7.2TW$. Notons néanmoins qu’une source dont l’intensité fluctue ne complexifie pas l’implémentation de la solution. Finalement, on sait que la voile extérieure contribue à 90% du poids total de la fusée, signifiant donc qu’à la séparation, la voile intérieure sera équivalente au dixième de la masse.

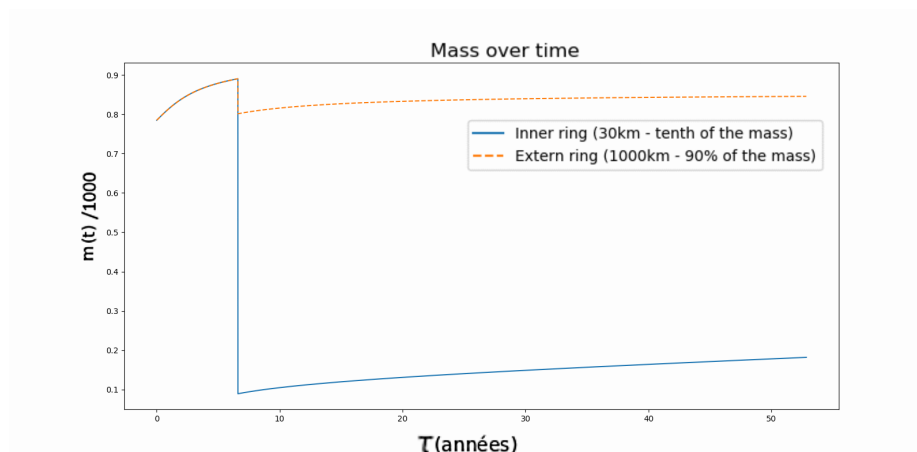


FIGURE 3.8 – Évolution de la masse inertielle des deux étages de la fusée par rapport au temps propre du voyageur suivant les valeurs utilisées dans le modèle de Forward.

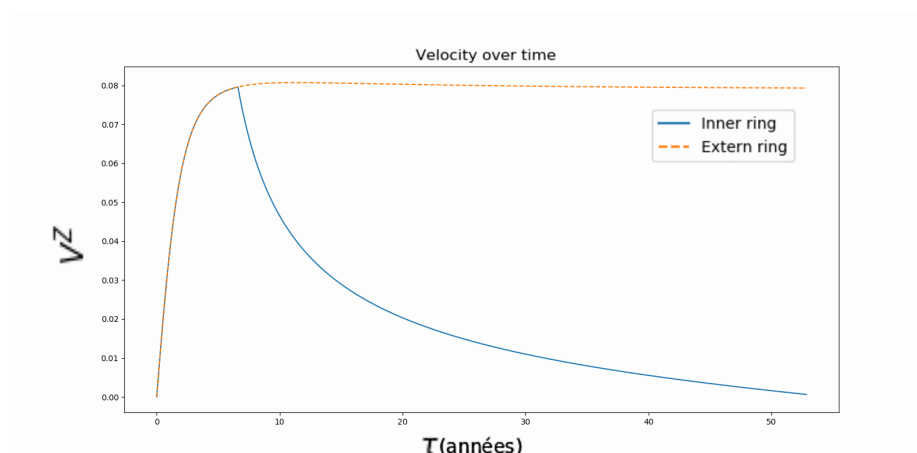


FIGURE 3.9 – Évolution de la 3-vitesse des deux étages de la fusée par rapport au temps propre du voyageur suivant les valeurs utilisées dans le modèle de Forward.

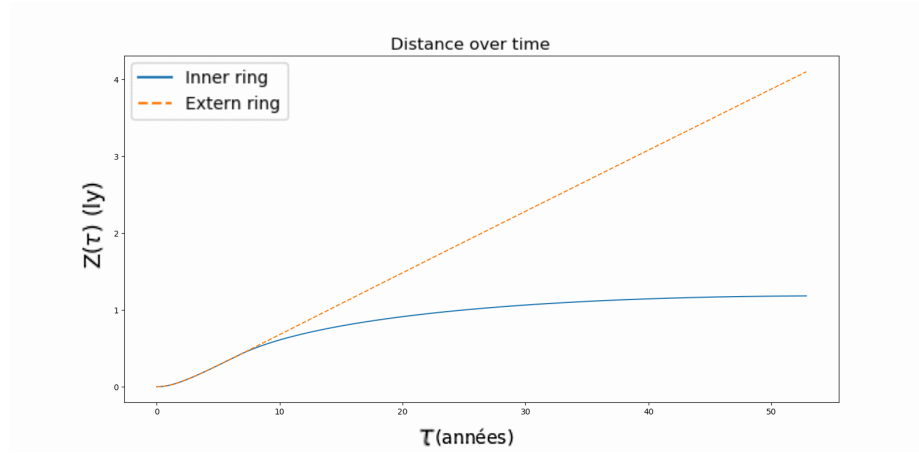


FIGURE 3.10 – Évolution de la distance parcourue par les deux étages de la fusée par rapport au temps propre du voyageur suivant les valeurs utilisées dans le modèle de Forward.

Comme on peut le remarquer, l'accélération fournie par le laser ne permet au payload de n'atteindre qu'une année-lumière, là où la voile extérieure arrive à atteindre les 4 années-lumière, au bout d'une cinquantaine d'années. Une des justifications que l'on peut donner à cette distance insuffisante pour atteindre Alpha Centauri est le fait que nous utilisons une source constante (7.2TW), là où Forward finit le trajet avec une source plus de 3 fois supérieure à celle initiale. De plus, comme nous le verrons plus bas, utiliser un autre modèle de diffusion de la lumière permettra d'atteindre cette distance plus facilement. Finalement, les échelles de grandeurs semblent être relativement semblables à celle de l'article de Forward, proposant un voyage de 41 années pour parcourir les 4.3 années-lumière nous séparant d'Alpha Centauri.

Finalement, nous pouvons jeter un coup d'oeil sur la quantité totale d'énergie demandée par un tel trajet (bien que ce dernier soit incomplet, comme vu ci-dessus) : au bout de 50 années pour le voyageur, la quantité totale d'énergie déployée aura été de l'ordre de $7.2\text{TW} * 50 \text{ années} = 1\text{e}22 \text{ J}$, ce qui représente une quantité colossale d'énergie.

3.3.3 Diffusion du laser

Tentons à présent de complexifier le modèle en rajoutant la notion de diffusion de laser, introduite précédemment à l'aide du facteur suivant (3.22) :

$$\left(\frac{D_{max}}{Z(\tau)}\right)^2 \quad (3.30)$$

Cette expression restera valide pour les deux voiles tant que la séparation n'a pas encore eu lieu. Une fois la séparation effectuée, on changera la distance parcourue par les photons de la manière suivante, pour la partie interne du vaisseau :

$$\left(\frac{D_{max}}{Z_e(\tau) + (Z_e(\tau) - Z_i(\tau))} \right)^2 \quad (3.31)$$

la fonction étant inchangée pour le module extérieur. Ainsi, l'expression 3.30 restera valide pour la voile extérieure, mais la distance parcourue pour atteindre la voile intérieure étant plus grande une fois la séparation effectuée, il faudra alors utiliser l'expression 3.31.

3.3.4 Effet Doppler

Lors de la séparation, il faudra reconsidérer notre fonction de puissance, la vitesse des deux voiles évoluant de manière particulièrement différente. On considérera donc le premier décalage de fréquence perçu par la voile externe en fonction de sa vitesse (introduit dans l'expression 3.24), sur lequel s'appliquera un second décalage en fonction de la différence de vitesse entre les voiles externe et interne. Cette vitesse relative des voiles extérieure et intérieure est donnée par :

$$V_{e,i}^Z = \frac{(V_i^Z - V_e^Z)}{1 - \frac{V_e^Z V_i^Z}{c^2}} \quad (3.32)$$

Le décalage Doppler appliqué entre les deux voiles sera donc exprimé ainsi :

$$\sqrt{\frac{1 + \frac{V_{e,i}^Z}{c}}{1 - \frac{V_{e,i}^Z}{c}}} \quad (3.33)$$

La fonction de puissance finale f^Z introduite dans les expressions 3.25 et 3.26 se verront dès lors modifiée, suite à l'ajout des modifications sur la diffusion du laser et l'effet Doppler. Après la séparation, la voile externe préservera le même modèle qu'avant la séparation, et la voile interne subira une fonction de puissance modifiée telle que :

$$f^Z = (1 + \epsilon) * \sqrt{\frac{1 + \tanh(\psi_e)}{1 - \tanh(\psi_e)}} * \sqrt{\frac{1 + \frac{V_{e,i}^Z}{c}}{1 - \frac{V_{e,i}^Z}{c}}} * P_{source} * 1 \quad (3.34)$$

si $Z_e(\tau) + (Z_e(\tau) - Z_i(\tau)) \leq D_{max_i}$, ou sinon :

$$f^Z = (1 + \epsilon) * \sqrt{\frac{1 + \tanh(\psi_e)}{1 - \tanh(\psi_e)}} * \sqrt{\frac{1 + \frac{V_{e,i}^Z}{c}}{1 - \frac{V_{e,i}^Z}{c}}} * P_{source} * \left(\frac{D_{max_i}}{Z_e(\tau) + (Z_e(\tau) - Z_i(\tau))} \right)^2 \quad (3.35)$$

Afin de jauger l'impact qu'aura l'effet Doppler sur un voyage de ce type, les figures 3.11, 3.12 et 3.13 comparent l'évolution d'une fusée impactée par l'effet Doppler à l'évolution d'une fusée qui n'est pas impactée par cet effet. Dans le cadre de ces intégrations, les mêmes valeurs initiales que celles de l'article de Forward sont prises, à la différence qu'on ne sépare à aucun moment les étages afin de se concentrer sur l'effet Doppler.

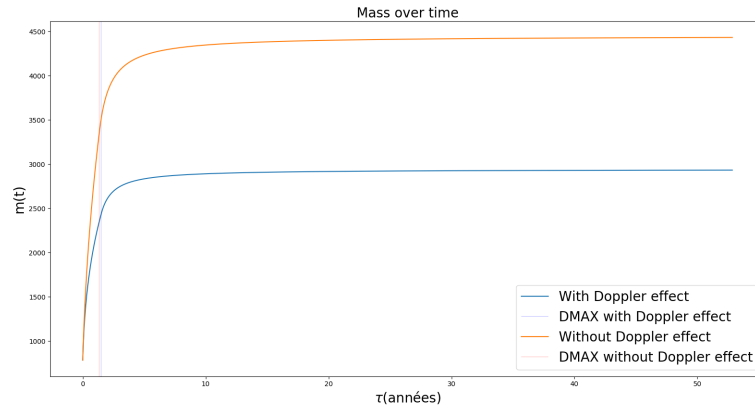


FIGURE 3.11 – Evolution de la masse inertielle par rapport au temps propre du voyageur, en différenciant la trajectoire soumise à l'effet Doppler ou non. DMAX correspond, pour chacune des deux trajectoires, à l'instant où le faisceau lumineux commence à perdre en efficacité.

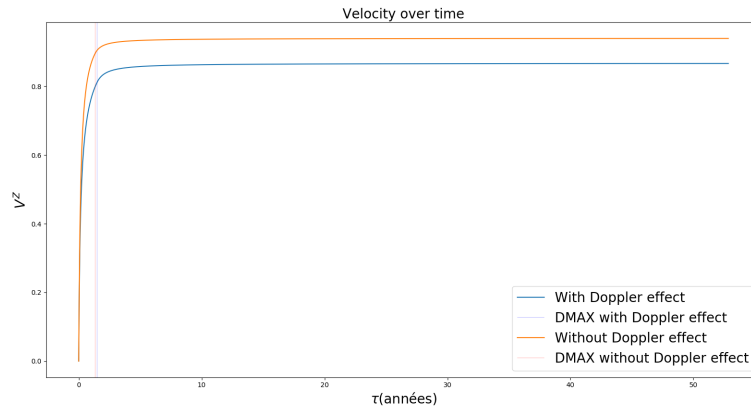


FIGURE 3.12 – Evolution de la vitesse de la fusée pour l'observateur inertiel par rapport au temps propre du voyageur, en différenciant la trajectoire soumise à l'effet Doppler ou non. DMAX correspond, pour chacune des deux trajectoires, à l'instant où le faisceau lumineux commence à perdre en efficacité.

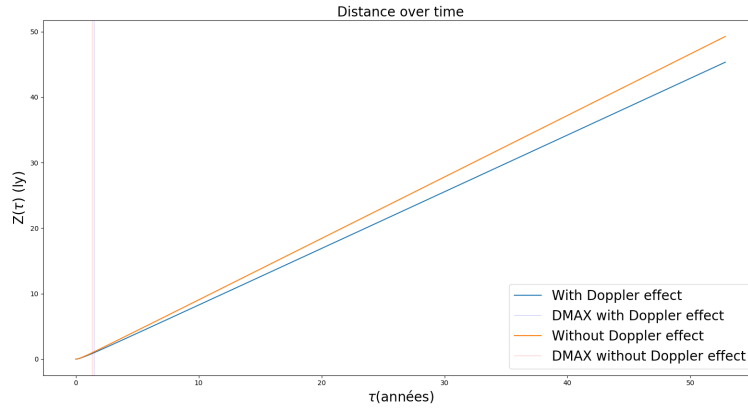


FIGURE 3.13 – Evolution de la distance parcourue pour l’observateur inertiel par rapport au temps propre du voyageur, en différenciant la trajectoire soumise à l’effet Doppler ou non. DMAX correspond, pour chacune des deux trajectoires, à l’instant où le faisceau lumineux commence à perdre en efficacité.

3.3.5 Résultats finaux

Possédant désormais une intégration prenant en compte un nombre significatif de paramètres influant sur l’évolution des voiles solaires, nous sommes désormais en mesure d’obtenir les résultats qui nous serviront de résultats finaux pour cette section, dans les figures 3.14, 3.15 et 3.16. A titre comparatif, on demeure sur les mêmes conditions initiales que celles proposées par Forward : on considère une fusée de 785 tonnes, propulsée par un laser d’une intensité constante de $7.2TW$ et où la voile extérieure contribue à 90% du poids total de la fusée. De plus, comme proposé dans l’article de Lubin, nous considérerons que le laser possède un diamètre de 10km et une longueur d’onde de 1064 nm.

Comme on peut l’observer, avec les conditions initiales proposées par Forward, la voile intérieure est en mesure d’arriver vers Alpha Centauri sans trop de marge d’erreur. A la différence avec l’article de Forward, la séparation se fait ici très tôt, le laser n’augmentant pas en intensité, et le freinage étant dès lors plus compliqué sur la fin que lors du début.

On peut également voir sur la figure 3.17 l’évolution du temps du voyageur embarqué par rapport à l’observateur inertiel, exprimant ainsi la dilatation du temps impliquée par l’accélération subie par le voyageur. [4]

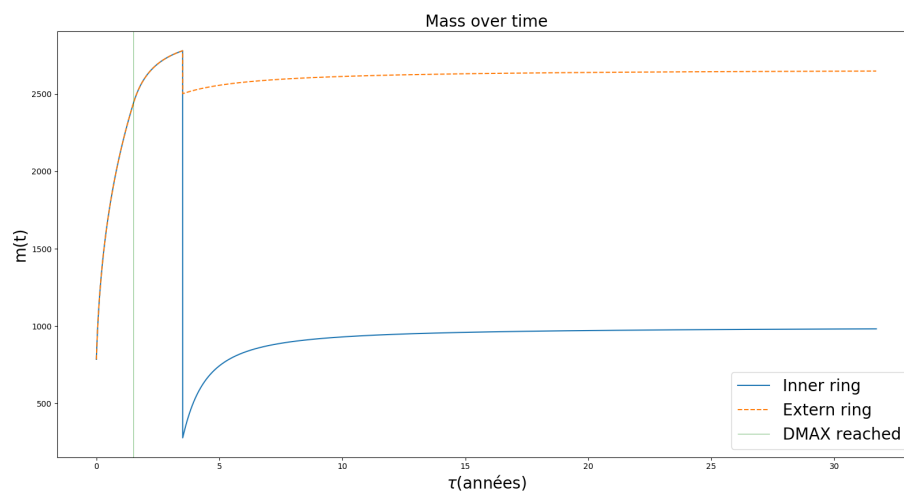


FIGURE 3.14 – Évolution de la masse inertielle par rapport au temps propre du voyageur, pour une voile solaire associée aux conditions initiales proposées par le modèle de Forward.

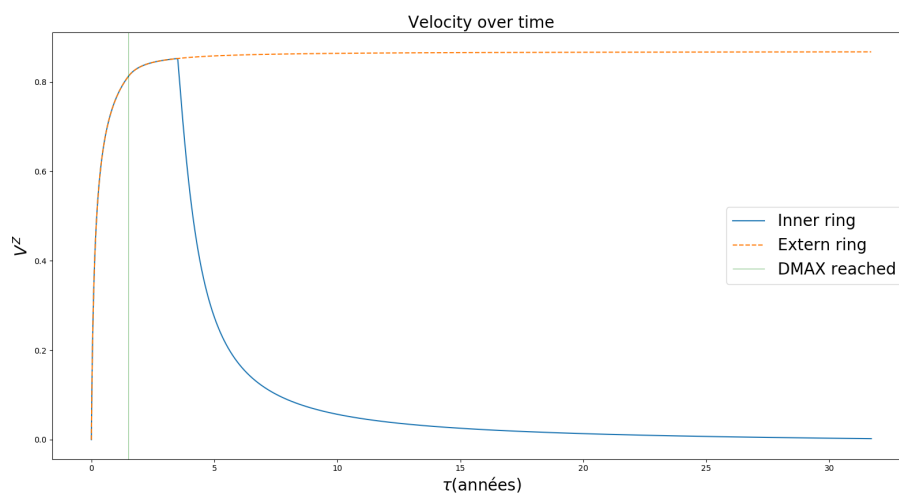


FIGURE 3.15 – Évolution de la vitesse par rapport au temps propre du voyageur, pour une voile solaire associée aux conditions initiales proposées par le modèle de Forward.

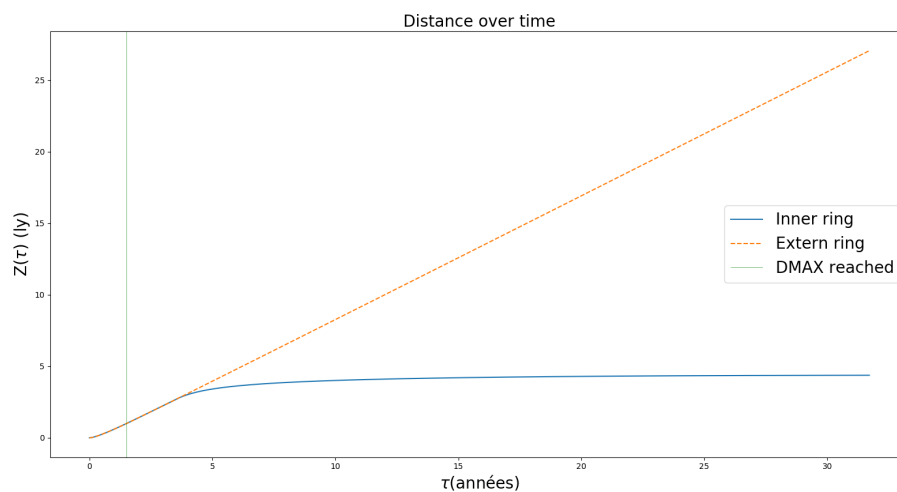


FIGURE 3.16 – Évolution de la distance par rapport au temps propre du voyageur, pour une voile solaire associée aux conditions initiales proposées par le modèle de Forward.

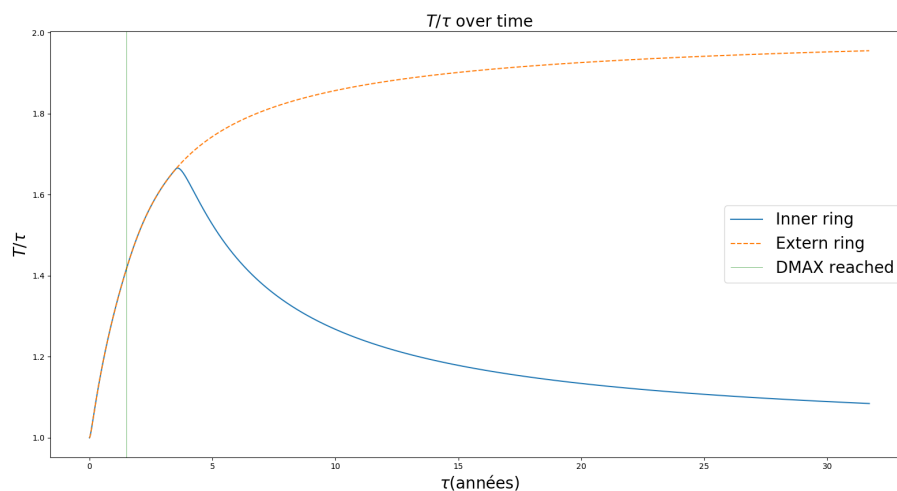


FIGURE 3.17 – Ratio entre la durée écoulée par l'observateur inertiel et le voyageur, exprimant la dilatation du temps subie par la fusée.

Afin de déterminer la moment idéal de la séparation des deux étages, un algorithme de recherche de minimum peut être appliqué sur notre intégrateur numérique. Cet algorithme minimisera la distance séparant la voile solaire de la destination une fois que la voile solaire sera à l'arrêt, en fonction de la distance à laquelle est effectuée la séparation 3.18.

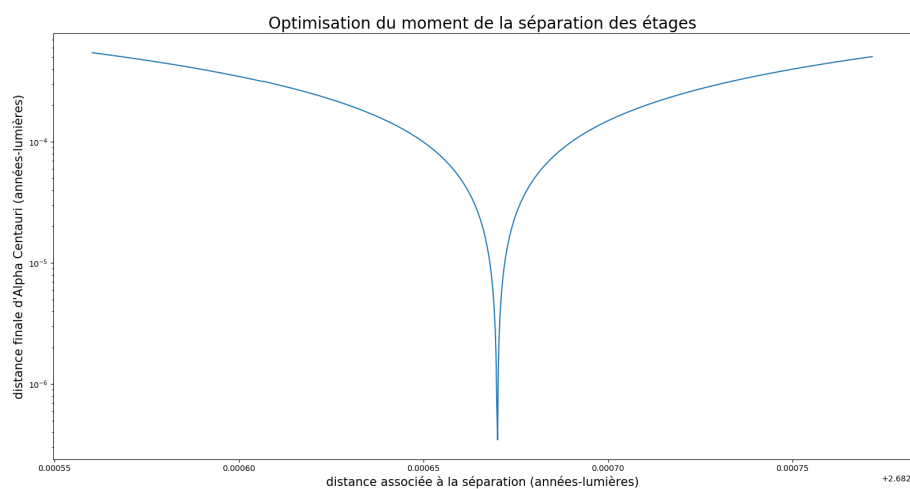


FIGURE 3.18 – Distance séparant la voile solaire d'Alpha Centauri une fois la voile à l'arrêt, en fonction de la distance à laquelle prend lieu la séparation des deux voiles.

Chapitre 4

Propagation de la lumière en direction de la voile solaire

4.1 Formalisme Hamiltonien

4.1.1 Obtention des équations géodésiques

A présent, nous souhaitons analyser les trajectoires des géodésiques lumières autour de la fusée, afin de pouvoir calculer la trajectoire des photons atteignant le voyageur embarqué. Pour cela, nous savons déjà que la géométrie de l'espace-temps autour de la fusée est dépendante de deux fonctions : la fonction M (3.4) associée à la masse de la fusée, ainsi que la fonction α (3.9) associée à la 4-accélération de la fusée, fonctions du temps propre τ .

La courbe \mathcal{G} de la géodésique sera dans notre analyse caractérisée par le système de coordonnées de la métrique entrante(ou sortante) 3.3, donnant lieu à des expressions de la forme :

$$\mathcal{G} = (c\tau(\sigma), r(\sigma), \theta(\sigma), \varphi(\sigma)) \quad (4.1)$$

où σ est un paramètre affine de la géodésique. Ces fonctions \mathcal{G} peuvent être obtenues en résolvant les équations des géodésiques, mais la manipulation de la métrique étant complexe dans ce cadre, l'utilisation d'une formulation hamiltonienne des géodésiques sera utilisée.

Dans cette optique, nous utiliserons le fait que les équations des géodésiques soient des équations d'Euler-Lagrange pour le Lagrangien suivant :

$$L = 1/2 g_{\mu\nu} \dot{x}^\mu \dot{x}^\nu \quad (4.2)$$

où $\dot{x}^\mu = \frac{dx^\mu}{d\sigma}$. En utilisant l'impulsion généralisée (moment conjugué) usuel $p_\alpha = \frac{\partial L}{\partial \dot{x}^\alpha}$, nous pouvons introduire le hamiltonien suivant :

$$H = \frac{1}{2} g^{\alpha\beta} p_\alpha p_\beta \quad (4.3)$$

De là, nous pouvons définir les expressions recherchées pour la dynamique des géodésiques à partir de deux groupes d'expressions :

$$\frac{dx^\mu}{d\sigma} = g^{\mu\nu} p_\nu \quad (4.4)$$

où $\{x^i\}_{i=0,1,2,3} = (u, r, \varphi, \theta)$ et

$$\frac{dp_\mu}{d\sigma} = -\frac{\partial g^{\alpha\beta}}{2\partial x^\mu} p_\alpha p_\beta \quad (4.5)$$

Ces expressions peuvent être développées à partir des composants $[g^{\alpha\beta}]$ de la métrique contra-variant décrits tels que :

$$[g^{\alpha\beta}] = \begin{pmatrix} 0 & \pm 1 & 0 & 0 \\ \pm 1 & 2r\alpha \cos(\theta) - 1 + 2M/r & -\alpha \sin(\theta) & 0 \\ 0 & -\alpha \sin(\theta) & -1/r^2 & 0 \\ 0 & 0 & 0 & -1/(r^2 \sin^2(\theta)) \end{pmatrix} \quad (4.6)$$

le signe \pm dépendant du fait que la métrique soit entrante ou sortante. Les deux groupes d'expressions définies au-dessus peuvent dès lors être exprimées de la manière suivante, proposant ainsi une manière d'obtenir les trajectoires des géodésiques (en assumant que $c = 1$) :

$$\frac{d\tau}{d\sigma} = \pm p_r \quad (4.7)$$

$$\frac{dr}{d\sigma} = \pm p_r - p_r \left(1 - 2\frac{M}{r} - 2\alpha r \cos(\theta) \right) - p_\theta \alpha \sin(\theta) \quad (4.8)$$

$$\frac{d\theta}{d\sigma} = -\frac{p_\theta}{r^2} - p_r \alpha \sin(\theta) \quad (4.9)$$

$$\frac{d\varphi}{d\sigma} = -\frac{p_\varphi}{r^2 \sin^2(\theta)} \quad (4.10)$$

$$\frac{dp_\tau}{d\sigma} = p_r p_\theta \sin(\theta) \dot{\alpha} - \frac{p_r^2}{r} (\dot{\alpha} \cos(\theta) r^2 + \dot{M}) \quad (4.11)$$

$$\frac{dp_r}{d\sigma} = \frac{p_r^2}{r^2} (-\alpha \cos(\theta) r^2 + M) - \frac{p_\varphi^2}{r^3 \sin^2(\theta)} - \frac{p_\theta^2}{r^3} \quad (4.12)$$

$$\frac{dp_\theta}{d\sigma} = p_r \alpha (p_\theta \cos(\theta) + p_r r \sin(\theta)) - \frac{p_\varphi^2 \cos(\theta)}{r^2 \sin^3(\theta)} \quad (4.13)$$

On remarquera que la métrique ne dépendant pas de φ , p_φ sera ici considéré comme une constante. L'hamiltonien associé, qui lui aussi sera constant, sera le suivant :

$$H = -\frac{p_\varphi^2}{2r^2 \sin^2(\theta)} - \frac{p_r^2}{2} \left(1 - \frac{2M}{r} - 2r\alpha \cos(\theta)\right) - \alpha \sin(\theta) p_r p_\theta \pm p_r p_r - \frac{p_\theta^2}{2r^2} \quad (4.14)$$

En fonction du type de géodésique souhaité, on pourra observer une valeur différente pour le hamiltonien. Pour les trajectoires de photons (géodésiques lumières), on considère un H nul. En revanche, si l'on souhaite considérer des géodésiques de matière, on utilisera les mêmes expressions 4.4 et 4.5, à la différence que l'on prendra soin de conserver un H constant avec $H < 0$.

Comme nous l'avons mentionné plus haut, la géométrie de l'espace-temps autour de la fusée est définie par les deux fonctions M et α , fonctions que l'on a déjà obtenues au cours de la section précédente, en résolvant les équations de fusées relativistes. On aura pour rappel que le temps propre τ correspond à la coordonnée u de l'équation 3.3, dans les limites où $M \rightarrow 0$ où $r \rightarrow \infty$.

Sur la figure 4.1, nous pouvons observer la manière par laquelle convergeront vers l'observateur les géodésiques lumières en fonction de leur angle de réception θ_R . On considérera donc un r nul et une valeur $\tanh(\psi_R) = 0.7$, c'est-à-dire que $V^Z(\tau_R) = 0.7c$, et où τ est exprimé comme fonction du paramètre affine σ de la trajectoire de la géodésique lumière. De plus, afin de valider l'intégration numérique, une analyse du Hamiltonien tout le long de celle-ci peut être effectuée, une valeur qui deviendrait trop élevée exprimant le fait que la simulation ne serait plus fiable 4.2.

4.1.2 Gestion de la masse dans le cadre des équations de géodésiques

On notera que la fonction M liée à la masse pourra être négligée dans le cas de luminosité très importante. Pour en arriver à cette déduction, nous ré-écrivons la métrique 3.3 en effectuant des changements de variables basées sur les paramètres introduits précédemment. En considérant $s = \tau/\tau_c$, et $R = r/(c\tau_c)$, avec τ_c l'échelle caractéristique du temps propre de la fusée $\tau_c = m_0 c^2/|f^T|$, on peut réduire le terme de masse M/r de la métrique à $\mathcal{M}/R \times (G|f^T|/c^5)$, avec $m = \mathcal{M}m_0$ et $\alpha M = -\frac{d\psi}{ds} \mathcal{M} \times (G|f^T|/c^5)$. Dès lors, en remplaçant $|f^T|$ par la luminosité L liée à la voile solaire, tant que $\frac{GL}{c^5} \ll 1$, la fonction M pourra être mise de côté dans le cadre de notre analyse.

4.2 Aberrations relativistes et effets Doppler

Dans le cadre de l'intégration des différentes équations de géodésiques introduites ci-dessus, il sera important d'introduire deux effets importants intervenant dans l'observation de la lumière

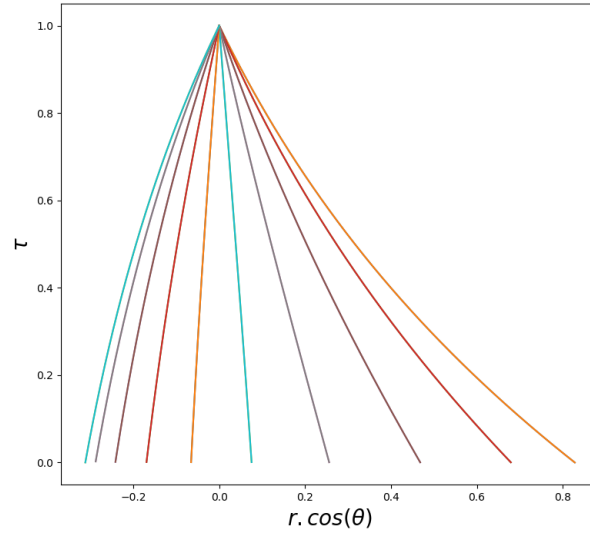


FIGURE 4.1 – Convergence des géodésiques lumières vers l’observateur accéléré pour $\tanh(\psi_R) = 0.7$ et différentes valeurs de θ_R .

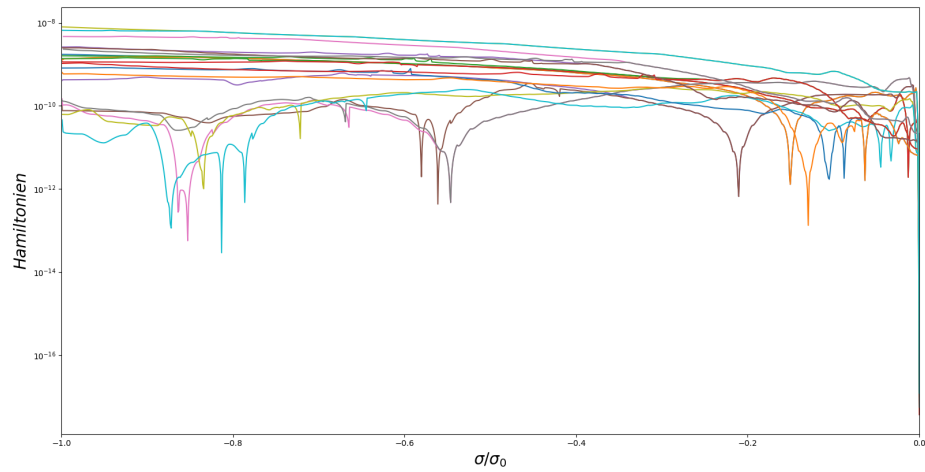


FIGURE 4.2 – Évolution du Hamiltonien

pour un voyageur associé à des vitesses relativistes. Le premier de ces effets est l'aberration relativiste, et le second, plus connu, est le décalage de fréquence qui pourra être observé (effet Doppler).

Par simplicité, nous considérerons que notre voyageur est associée à une trajectoire accélérée sur un axe, commençant son périple au repos en $\tau = 0$ avec $Z(0) = T(0) = \psi(0) = 0$, où τ sera son temps propre et où les coordonnées sphériques (r, θ, φ) lui seront associées, afin d'établir sa sphère céleste. Le voyageur commençant au repos en $\tau = 0$, la sphère céleste initiale sera identique à celle de l'observateur inertiel. Pour rappel, comme établi lors du chapitre précédent, l'observateur inertiel est l'observateur dont la ligne d'univers sera tangente à celle du voyageur lors du départ en $\tau = 0$, et qui demeurera au repos.

Ne faisant pas utilisation de luminosité extrêmement grande dans le cadre de la propulsion de la fusée, nous négligerons donc le paramètre de masse en considérant que $M = 0$. La masse étant considérée comme nulle, l'espace-temps pour le voyageur accéléré semblera plat, ce qui nous ramènera à la métrique de Minkowski pour la perception de l'espace-temps pour ce dernier. Les effets de déviation angulaires subits en conséquence à l'accélération impliqueront l'apparition d'aberration et de décalage de fréquence différents de ceux observés en relativité restreinte à vitesse constante. Ces effets seront d'une importance capitale dans le cadre d'une bonne réception de télécommunications ou d'une navigation basée sur la position des étoiles perçues.

4.2.1 Analyse des aberrations relativistes

Nous avons vu que les trajectoires des rayons lumineux peuvent être obtenus en imposant un Hamiltonien nul, et en solutionnant les expressions 4.7–4.13, solution uniquement dépendante de la fonction d'accélération $\alpha(\tau)$, maintenant le M négligé. Dans l'optique de l'obtention de ces trajectoires, nous intégrerons numériquement les équations en remontant le temps, depuis le moment de la réception ($\tau = \tau_R$, $r = 0$, $\theta = \theta_R$, $\varphi = \varphi_R$) jusqu'au moment de l'émission des photons, depuis la sphère de référence ($\tau = 0$, $r = r_E$, $\theta = \theta_E$, $\varphi = \varphi_E$).

Les trajectoires dont nous aurons besoin afin de recréer la sphère initiale du voyageur accéléré (en $r = 0$) répondront dans un premier temps au fait que $\varphi_E = \varphi_R$, impliqué par $p_\varphi = 0$ constant (symétrie axiale). Ensuite, la réception d'un photon en $r = 0$ implique que la courbe de cette trajectoire respecte $p_\theta(\tau_R) = 0$.

Il ne nous reste plus qu'à définir les deux paramètres $p_r(\tau_R)$ et $p_\tau(\tau_R)$, les paramètres θ_R et τ_R étant considérés comme paramètres libres. L'équation 4.7 nous permet de dire que si on impose $p_r(\tau_R) = 1$, le comportement de τ sera similaire à celui de σ lors de la réception. Étant donné que l'on sait déjà que le Hamiltonien devra être nul, et que tous les termes du Hamiltonien sont déjà connus à part le paramètre $p_\tau(\tau_R)$, on peut retrouver ce dernier paramètre en résolvant le Hamiltonien.

Une fois ces paramètres trouvés, nous aurons en notre possession toutes les conditions initiales permettant d'intégrer en remontant le temps les différentes équations de géodésiques, jusqu'à ce que l'on atteigne $\tau = 0$, correspondant à l'émission des différents photons depuis la sphère céleste de référence.

L'aberration relativiste pourra être observée par un changement de positions des différents objets visibles sur la sphère céleste, en comparant la sphère céleste de référence et celle lors de la réception des photons. On sait que $\varphi(\tau)$ restera constant, c'est dès lors dans la différence entre θ_E et θ_R que l'aberration se manifestera pour les géodésiques lumières. Un second facteur contribuant à cette aberration est le fait que la coordonnée angulaire θ_R ne correspondra plus à celle souhaitée en $\tau = \tau_R$ avec $\psi_R \neq 0$.

On introduira dès lors l'angle Θ_R qui correspond donc à l'angle souhaité sur la sphère céleste locale du voyageur. Cet angle peut être obtenu en revenant à la sphère céleste du voyageur en le considérant au repos. Cela peut être obtenu en imposant :

$$Z_R - Z_{\mathcal{L}}(\tau_R) = \rho \cos \Theta_R \quad (4.15)$$

et

$$X^2 + Y^2 = r_R^2 \sin^2(\theta_R) = \rho^2 \sin^2(\Theta_R) \quad (4.16)$$

en utilisant les transformations de coordonnées 3.6, et où ρ et Θ_R correspondent aux coordonnées sphériques de la sphère locale (respectivement associées à r et θ).

On voulait initialement déterminer la relation liant Θ_R et θ_R lorsque $\tau = \tau_R$, relation qui peut désormais être exprimée de la manière suivante :

$$\tan \Theta_R(\beta + \cos \theta_R) = \sin \theta_R(1 - \beta^2)^{1/2} \quad (4.17)$$

avec $\beta = \tan \psi_R$, étant donné que l'on a, pour rappel, déjà établi le profil de ψ lors du chapitre précédent.

On remarquera que cette dernière expression est identique à celle proposée par Einstein en relativité restreinte, pour l'aberration relativiste associée à une vitesse constante, où θ_R devient alors l'angle d'incidence mesurée par l'observateur au repos (avec $\theta_E = \theta_R$ en relativité restreinte). L'aberration pourra finalement être obtenue, étant égale à $\Theta_R(\theta_E)$, ou pour rappel θ_E est obtenu en intégrant les équations de géodésiques en remontant le temps, à partir de θ_R choisi arbitrairement (composant avec φ_R la position dans la sphère céleste locale du voyageur vers laquelle nous souhaitons porter notre regard). Ainsi, on pourra avec la figure 4.3 observer les correspondances entre θ_E et θ_R pour différentes vitesses.

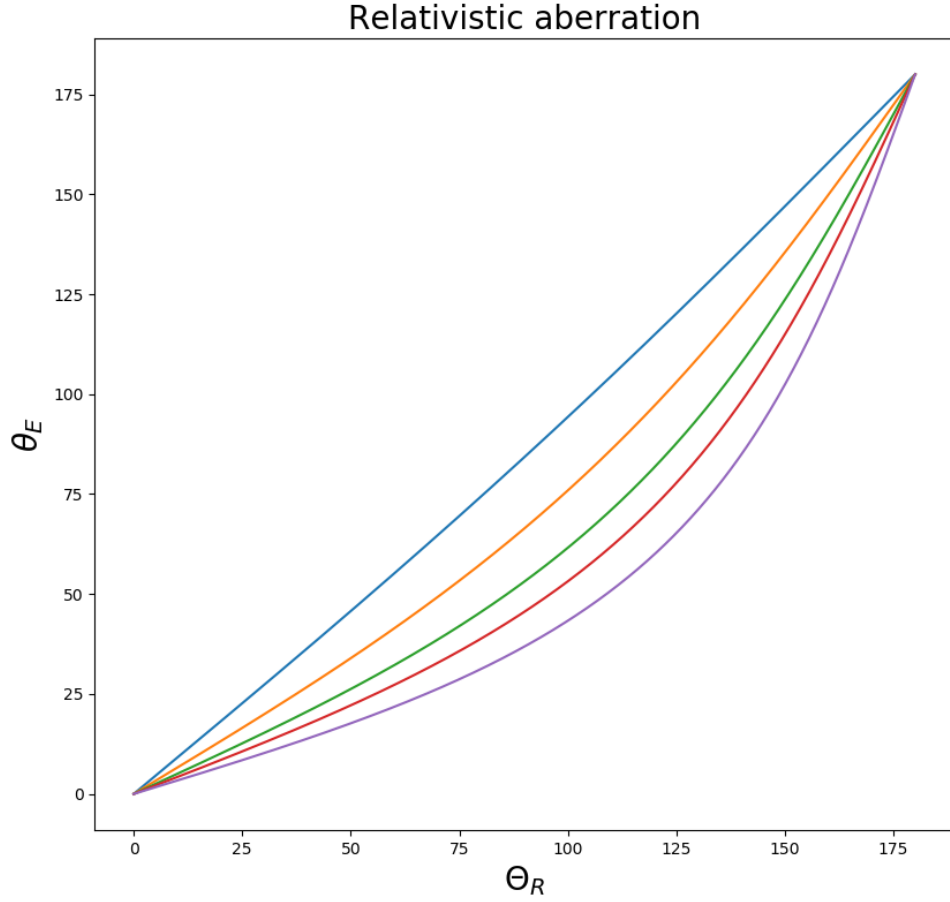


FIGURE 4.3 – Aberrations relativistes observées pour différentes vitesses ($V^Z/c = 0.1, 0.4, 0.6, 0.8, 0.95$). L'angle Θ_R correspondant à l'observation du voyageur sera exprimé en fonction de l'angle θ_E correspondant à l'angle d'émission du signal lumineux.

4.2.2 Analyse de l'effet Doppler

Maintenant la problématique de l'aberration relativiste traitée, nous pouvons aborder le problème du décalage de fréquence apparaissant sur les rayons lumineux perçus par le voyageur accéléré.

Pour aborder cet effet, définissons dans un premier temps l'expression mesurant l'énergie d'un photon observée par un observateur lors d'un événement espace-temps e :

$$E_e = h\nu_e = (p_\mu \lambda'^\mu)_e \quad (4.18)$$

où λ'^μ est le vecteur unité tangent à la ligne d'univers de l'observateur \mathcal{O}' , h est la constante de Planck, et où ν_e est la fréquence mesurée du photon à l'événement e . On considère ici que celui qui reçoit le signal lumineux est le voyageur accéléré, dont la ligne d'univers associée est définie comme $(r = 0, \tau)$ dans ses coordonnées locales. La fréquence du signal reçu ν_R sera alors proportionnelle à $p_\tau(\tau_R)$, qui était pour rappel obtenu à l'aide du Hamiltonien.

Nous considérerons que la source du signal lumineux en question, émise en $\tau = 0$, est au repos, et sera associée à une ligne d'univers $(X_E, Y_E, Z_E) = (X, Y, Z)_{\tau=0}$ ainsi qu'au temps propre T . Cette source lumineuse, pourra ici être associée à une étoile dont la lumière sera donc perçue par le voyageur accéléré, cette étoile étant alors associée aux positions (θ_E, φ_E) sur la sphère céleste de référence. On considérera également que la fréquence des signaux lumineux émis sera égale à celle observée par l'observateur inertiel, au repos sur Terre.

Les composants du vecteur tangent de la ligne d'univers d'une source lumineuse (une potentielle étoile dans notre situation) pourront être exprimés de la manière suivante :

$$dx^\mu/dT = d(\tau_E, r_E, \theta_E, \varphi_E)/dT \quad (4.19)$$

on où considère que $d\varphi_E/dT = 0$. Revenons un peu plus haut, où nous avons introduit les transformations 3.6. Dans ces expressions, le couple $(T_{\mathcal{L}}(\tau), Z_L(\tau))$ était associé aux fonctions spécifiant la ligne d'univers \mathcal{L} pour le voyageur dans les coordonnées inertielles de l'observateur \mathcal{O} .

Les expressions 3.6 peuvent donc être empruntées afin de calculer de la manière suivante les composants cités ci-dessus :

$$T_E = T_{\mathcal{L}} + r_E [\cosh(\psi_E) + \cos(\theta_E) \sinh(\psi_E)] \quad (4.20)$$

$$Z_E = Z_{\mathcal{L}} + r_E [\sinh(\psi_E) + \cos(\theta_E) \cosh(\psi_E)] \quad (4.21)$$

$$(X_E + Y_E)^{1/2} = r_E \sin(\theta_E) \quad (4.22)$$

L'équation 4.22 pourra avoir ses deux membres dérivés de la manière suivante, en fonction de T :

$$\frac{d\theta_E}{dT} = -\frac{\tan(\theta_E)}{r_E} \frac{dr_E}{dT} \quad (4.23)$$

Afin de dériver les deux autres expressions 4.20 et 4.21 en fonction de T , nous emploierons ce dernier résultat nouvellement acquis, que nous substituerons dans les différentes équations. Cela effectué avec l'aide des expressions 3.7 et 3.8, nous obtenons un nouveau système d'équations linéaires dont les inconnues seront $\frac{d\tau_E}{dT}$ et $\frac{dr_E}{dT}$, système dont la résolution pourra être effectuée par le biais des expressions suivantes :

$$\frac{d\tau_E}{dT} = \cosh \psi_E + \cos \theta_E \sinh \psi_E \quad (4.24)$$

$$\frac{dr_E}{dT} = -\cos \theta_E \sinh \psi_E + \alpha_E r_E \cos \theta_E (\cosh \psi_E + \cos \theta_E \sinh \psi_E) \quad (4.25)$$

Ce résultat nous permettra de retrouver la formulation du décalage Doppler en relativité restreinte, à l'aide de l'expression 4.18 :

$$E_E = p_\tau(0) \frac{d\tau_E}{dT} + p_r(0) \frac{dr_E}{dT} = \frac{1}{2} (-\cos \theta_E \sinh \psi_E + \cosh \psi_E) \quad (4.26)$$

en considérant que $p_\tau = \frac{1}{2}$, que $p_r = 1$ et que p_θ , p_φ et α sont tous les trois nuls. Comme $\frac{d\tau_R}{dT} = 1$ et $\frac{dr_R}{dT} = 0$, on aura également

$$E_R = p_\tau(\tau_R) \frac{d\tau_R}{dT} + p_r(\tau_R) \frac{dr_R}{dT} = \frac{1}{2} \quad (4.27)$$

Le ratio $\frac{E_E}{E_R}$ correspondra alors à l'effet Doppler relativiste souhaité, pour une source lumineuse non accélérée :

$$\left(\frac{\nu_R}{\nu_E} \right)_{SR} = (1 - \beta_E \cos \theta_E)^{-1} \sqrt{1 - \beta_E^2} \quad (4.28)$$

avec $\beta_E = \tanh \psi_E$.

En revanche, il peut être pertinent de pousser un peu le modèle pour voir comment se comporte cet effet si la source lumineuse est accélérée, autrement dit si α_E est non nul. On considérera par simplicité que la source lumineuse commence au repos, avec une vitesse ψ_E donc égale à zéro.

Dès lors, si on revient sur le système proposé par les trois expressions 4.23, 4.24 et 4.25, on obtient alors le système suivant :

$$\frac{d\tau_E}{dT} = 1 \quad (4.29)$$

$$\frac{dr_E}{d\tau} = r_E \cos(\theta_E) \alpha_E \quad (4.30)$$

$$\frac{d\theta_E}{d\tau} = -\sin(\theta_E) \alpha_E \quad (4.31)$$

On peut alors calculer le vecteur unité tangent de la source lumineuse, au moment de l'émission du signal lumineux, à l'aide de ces trois dernières relations :

$$\lambda'^{\tau,r,\theta} = \frac{1}{N} \frac{d(\tau_E, r_E, \theta_E)}{dT} \quad (4.32)$$

avec la norme du vecteur tangent $\lambda'^{\mu} = \frac{dx^{\mu}}{dT}$ qui sera exprimée de la manière suivante :

$$N = \sqrt{g_{\alpha\beta} \frac{dx^{\alpha}}{dT} \frac{dx^{\beta}}{dT}} \quad (4.33)$$

En résumé, comme précisé plus haut, le décalage Doppler sera exprimé comme le ratio entre l'énergie du photon à la réception E_R et à l'émission E_E , soit le ratio entre la fréquence reçue et émise, soit respectivement ν_R et ν_E . Une brève visualisation de ce décalage peut être effectué à l'aide de la figure 4.4, nous permettant d'observer les différents décalage issus du ratio (ν_R/ν_E) , pour différentes vitesses, en fonction de la valeur de l'angle Θ_R .

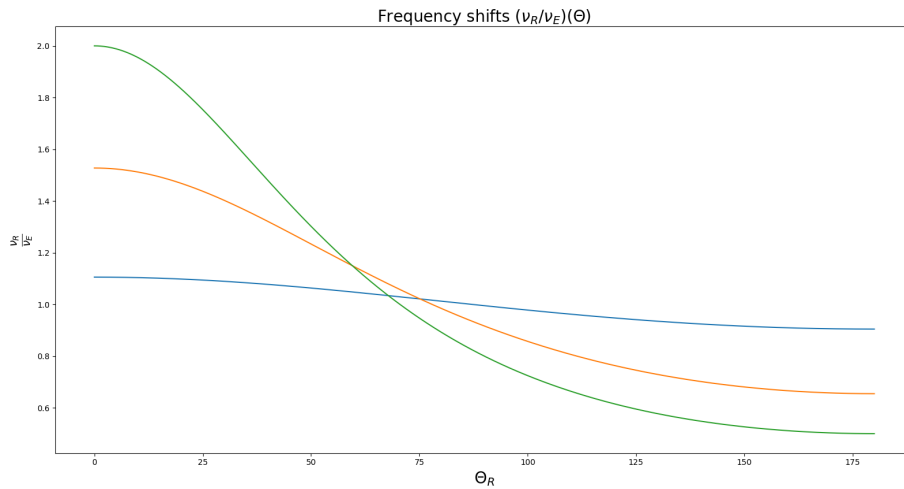


FIGURE 4.4 – Décalages de fréquences $(\nu_R/\nu_E)(\Theta)$ observés pour différentes vitesses ($V^Z/c = 0.1, 0.4, 0.6$), en fonction des différents angles Θ_R . Une valeur inférieure à 1 impliquera un décalage vers le rouge, une valeur supérieure à 1 impliquera un décalage vers le bleu.

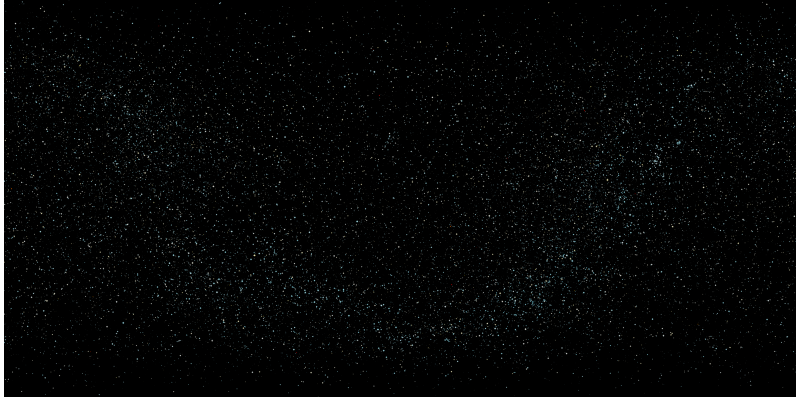


FIGURE 4.5 – Représentation d’une portion du Yale bright star catalogue[1]. Une limitation sur la magnitude des étoiles a été fixée à une magnitude de 8, afin de préserver un visuel clair.

4.3 Applications

Tentons à présent de construire un modèle de sphère céleste, afin d’avoir une idée plus concrète de ce que serait susceptible de voir un voyageur accéléré. Reprenons dès lors le cas d’une voile solaire à deux étages.

Dans un premier temps, il nous faut considérer une sphère céleste de référence. Pour ce faire, nous utiliserons le "Yale Bright Star Catalogues"[1], afin de dresser une première cartographie du ciel de référence, visible sur la figure 4.5. Dans le cadre de cette visualisation, je limite la magnitude apparente des étoiles à une magnitude de 8, afin d’éviter de surcharger l’affichage. Les couleurs sont choisies de manière assez grossière, à partir d’une estimation du type spectral des différentes étoiles. Ce type spectral peut être déduit de la température de l’étoile, elle-même déduite de la magnitude issue du système photométrique UBV (qui permet de classer les étoiles à partir de leurs magnitudes ultraviolette, bleue, et visible).

Une fois cela effectué, nous modifierons les coordonnées vis-à-vis de l’affichage afin d’avoir un visuel beaucoup plus intuitif à lire, sur la figure 4.6. L’idée de cette représentation est de visualiser la sphère céleste de référence de telle manière à ce que le centre du cercle représente les éléments associés à l’angle $\theta = 0$, et la bordure du cercle est associée à $\theta = \pi$.

Jusqu’avant la séparation, la voile solaire atteindra une vitesse avoisinant $V^Z/c = 0.85$. En appliquant pour chaque étoile (en considérant l’image pixel par pixel) trouvée depuis un angle θ_E une recherche afin d’obtenir l’angle Θ_R , nous saurons exactement là où se situera l’étoile sur la sphère céleste du voyageur accéléré en $\tau = \tau_R$. De la même manière, nous pourrions observer un décalage de fréquence Doppler appliqué sur ces étoiles, que l’on calculera à partir de la valeur RGB définie depuis le spectre de l’étoile. Juste avant la séparation, la figure 4.7 représente la sphère céleste locale du voyageur.

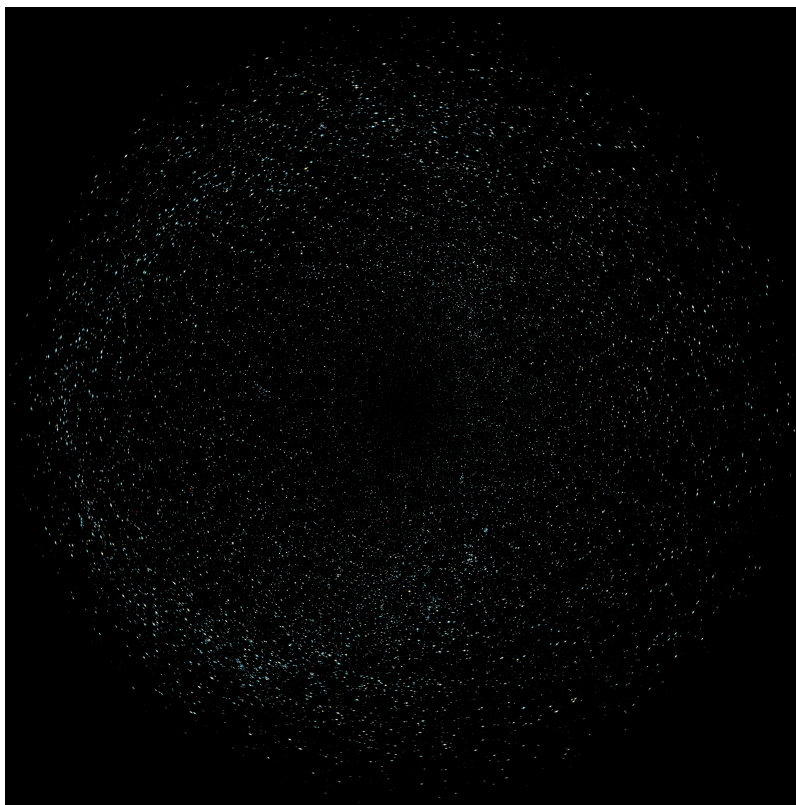


FIGURE 4.6 – Représentation de la sphère céleste de référence. Au centre de l'image se situe la direction vers laquelle l'observateur tourne le regard, et les éléments en bordure du cercle correspondant aux éléments étant dans son dos. Ainsi, sur le cercle dont le rayon est la moitié du rayon du grand cercle, nous aurons tous les éléments se situant à 90 degrés du regard de l'observateur (en haut, à droite, etc....)

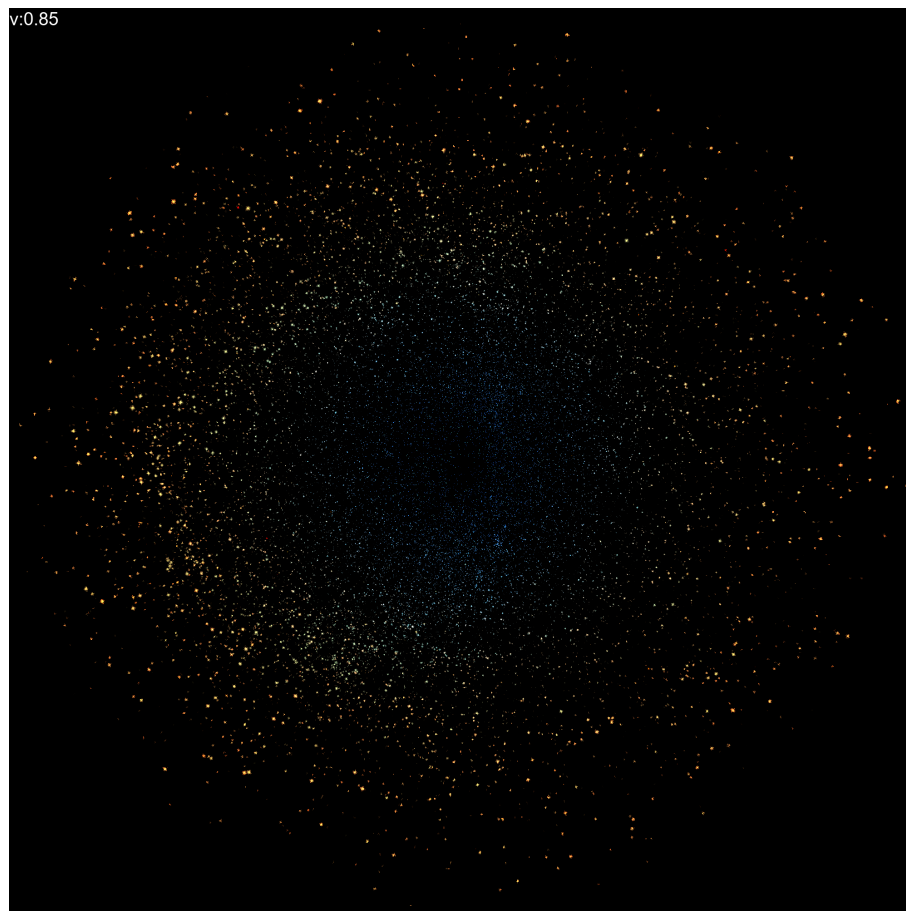


FIGURE 4.7 – Représentation de la sphère céleste locale du voyageur à bord de la voile solaire, peu avant la séparation, lorsque $\tanh(\psi_R) = 0.85$.

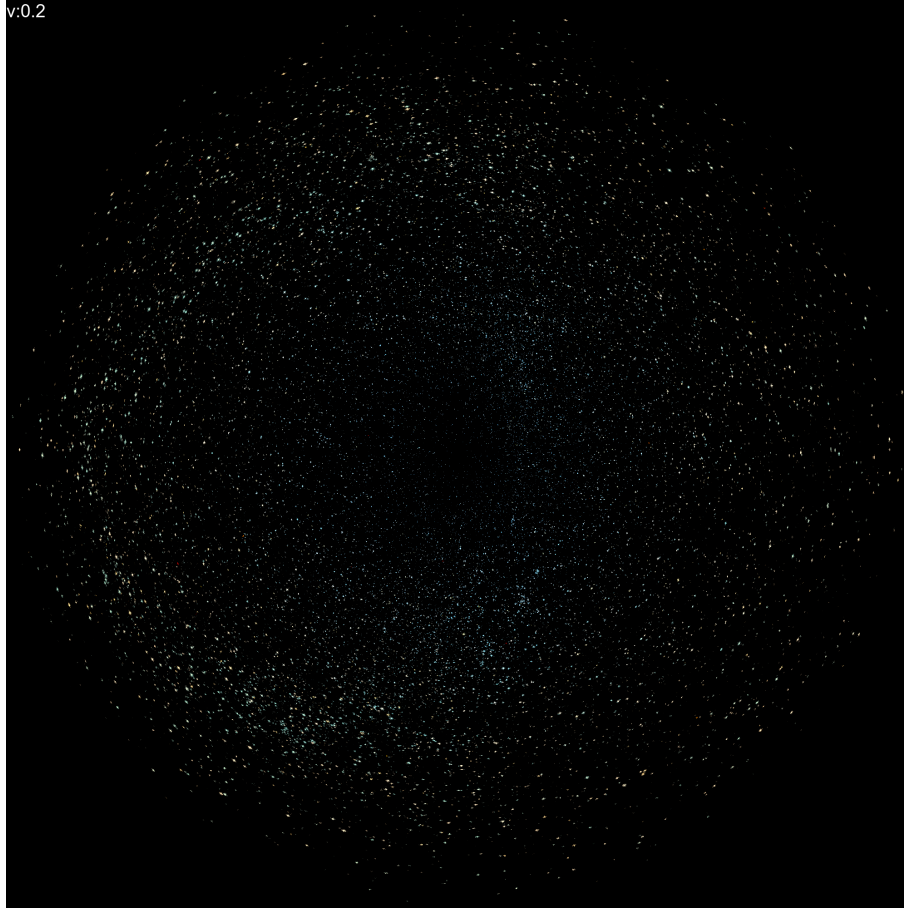


FIGURE 4.8 – Représentation de la sphère céleste locale du voyageur à bord de la voile solaire, plus ou moins deux ans après la séparation, lorsque $\tanh(\psi_R) = 0.2$.

Une fois la séparation effectuée, le voyageur subira une décélération, jusqu'à être à l'arrêt dans le système souhaité. A peu près deux ans après la séparation, la voile dans laquelle l'observateur voyage aura une vitesse de $V^Z/c = 0.2$. La figure 4.8 représente ce qu'il est alors susceptible d'observer.

La perte de masse suite à la séparation n'influera nullement sur les observations effectuées, cette dernière pouvant être négligée. Dès lors, l'unique paramètre régissant la modélisation de ces visualisations sera la fonction d'accélération $\alpha(\tau)$.

Chapitre 5

Conclusion

Nous avons pu aborder sous divers aspects la problématique d'un voyage interstellaire effectué à des vitesses relativistes, c'est-à-dire proche de celle de la lumière. Cette approche fut effectuée en considérant un modèle de voile solaire, caractérisé par un modèle de fusée dénuée de carburant massif.

Dans un premier temps, suite à une brève introduction à la problématique par le biais du problème du voyageur de Langevin, nous avons étudié un modèle simple, où nous souhaitions observer comment évoluait la trajectoire, la masse et le temps propre d'un voyageur accéléré. Nous avons pu remarquer que le temps propre du voyageur était ralenti, par rapport à un observateur inertiel resté au repos sur Terre. Ceci fut expliqué par le fait qu'un observateur accéléré peut percevoir cette accélération comme un champs de gravité local.

Nous avons ensuite étendu notre modèle à celui proposé par Forward, afin d'avoir en notre possession un modèle qui permettrait à une fusée non seulement d'atteindre un système cible, mais en plus de s'y arrêter, en faisant l'usage d'un second étage (une seconde voile), permettant ainsi de décélérer à l'approche du système. Cette approche fut accompagnée d'une analyse approfondie de l'effet Doppler appliqué sur le laser faisant office de source de propulsion, ainsi que de l'effet lié à la diffusion du laser à mesure que sa distance voyagee augmente.

Dans un second temps, nous nous sommes attardés sur le problème des géodésiques lumières atteignant un voyageur accéléré. La motivation principale de la réalisation de cette section décline du désir de savoir ce que verrait pratiquement un observateur situé à bord d'une voile solaire voyageant à des vitesses relativistes. Pour ce faire, nous avons employé la solution de Kinnorsley en relativité générale. Par le biais d'une formulation Hamiltonienne des équations des géodésiques, nous sommes parvenus à calculer les trajectoires des rayons lumineux atteignant notre voile solaire. Nous avons également étudié les effets observés sur l'aberration relativistes et le décalage Doppler témoignés sur ces signaux lumineux, nous permettant d'avoir une solution plus complète que celle proposée pour la relativité restreinte, dont les expressions ne sont ici valides que pour une vitesse constante. Dans ce cadre, nous avons pris

la liberté de négliger la masse de la fusée, car nous avons remarqué que cette dernière n'avait un impact conséquent que lors de l'usage de luminosité entrante extrêmement grande.

Bien que le voyage interstellaire demeure quelque chose d'utopique pour la vaste majorité de la population, ce travail nous montre que pratiquement, une telle chose n'est pas in-envisageable, dans un futur plus ou moins lointain, l'unique réelle difficulté majeure étant l'énergie requise pour garder actif un laser de cette puissance suffisamment longtemps. Mais qui sait si un jour, une telle énergie sera à notre portée, et que ces voyages seront alors éventuellement d'actualité.

Chapitre 6

Stage

6.1 Déroulement du stage

Dans ce chapitre, nous aborderons les différents points abordés lors de mes 3 mois de stage au CURL (Cosmology, Universe and Relativity at Louvain), situé au cyclotron de Louvain-La-Neuve. Mes trois mois de stage furent supervisés par deux maîtres de stage : Christophe Ringeval et Sébastien Clesse. Pour chaque maître de stage, divers objectifs pour les trois mois furent initialement établis.

Mon parcours universitaire étant majoritairement informatique, il fut décidé de tenter d'exploiter au mieux les compétences dont je disposais jusqu'alors, me faisant par conséquent travailler sur les supports informatiques des différents projets sur lesquels Christophe et Sébastien travaillaient.

L'objectif principal du stage fut associé au projet proposé par Christophe Ringeval. A l'aide d'une simulation, appelée ICARUS (Inhomogeneous Cosmology And Relativistic Universe Simulation), consistant en une simulation d'univers hétérogène, de multiples données avaient été produites, lesquelles avaient été traitées afin d'explorer la théorie des cordes cosmiques. De ces données ont été extraites diverses cordes cosmiques hypothétiques, ainsi que leur évolution au cours de la simulation. Chacune de ces cordes était alors caractérisée par une multitude de points les formant, un identifiant, une variable temporelle d'où pouvait être déduit l'âge de la corde, et finalement la longueur de la corde.

Ces données sous forme numérique étant difficiles à interpréter, l'idée de mon séjour au cyclotron était de me confier la création d'une simulation 3D dans laquelle il serait possible de visualiser ces cordes cosmiques. Le premier objectif était donc, à partir d'un fichier de données associées à un temps t de la simulation, d'afficher les différentes cordes cosmiques au temps t à l'aide de l'outil informatique de mon choix. Une fois cela fait, divers objectifs se rajouteraient à mesure que le projet avance.

En parallèle, dans le cadre du projet de Sébastien Clesse (et non plus de Christophe Ringeval), divers plus petits objectifs furent posés, s'étalant sur la durée du stage. Ces objectifs étaient tous axés autour du code ICARUS brièvement décrit ci-dessus, et de son efficacité. Il me fut demandé de préparer et de synthétiser l'usage d'un outil de documentation automatique pour le code ICARUS, écrit en Fortran. L'outil employé fut Doxygen, un outil de documentation automatique qui se prêtait bien à la situation. Après un bref survol de l'outil, j'ai donc effectué un compte rendu de la syntaxe à adopter pour son utilisation dans le cadre des différents scripts Fortran à documenter. Ensuite, nous dûmes travailler à deux, avec Sébastien Clesse, sur une parallélisation éventuelle de l'ensemble du code ICARUS, dans un premier temps sur CPU, et ensuite sur GPU.

6.2 Optimisation du code ICARUS

6.2.1 Introduction

Outre la documentation, une grande partie du stage, concernant le code ICARUS, consista donc en l'exploration des diverses alternatives que nous disposions pour paralléliser le code sur processeurs, et en l'exécution de nombreuses analyses effectuées sur des codes tests afin de déduire quelle syntaxe était la plus adaptée à adopter, dans le cadre de l'usage que l'on en faisait. Une fois le code parallélisé sur CPU, et un gain de performance conséquent obtenu (une réduction de temps dépassant un facteur 10, ce qui était on ne peut plus bénéfique, l'exécution du temps en séquentiel pouvant prendre plusieurs heures/jours, en fonction de sa configuration), on répéta la même opération de recherche pour la parallélisation sur carte graphique, car de multiples opérations matricielles étaient présentes, et l'usage de GPU étant préférable dans ce type de cas. Cette seconde recherche, accompagnée de ses tests sur une multitude de codes, fut plus rude que la première, de part le manque de documentation et d'accès vis-à-vis des outils recherchés (à savoir openACC pour Fortran, sur cluster), mais fut finalement fructueuse.

6.2.2 Documentation du code

Sous la demande de Sébastien Clesse, j'étais en charge de trouver un moyen de permettre une documentation automatique du code ICARUS, générant par conséquent des pages html permettant de naviguer à travers une documentation complète du projet. Dans ce cadre, je me suis tourné vers l'outil "Doxygen" permettant d'effectuer ce genre de tâche au travers du langage Fortran.

Ainsi, après avoir effectué quelques tests de mon côté afin d'être à l'aise avec l'outil en question, j'ai créé un bref fichier résumant l'utilisation de Doxygen dans le cadre du projet, en

présentant les différentes fonctionnalités que Sébastien Clesse avait jugées utiles de posséder.

6.2.3 Optimisation sur CPU

Le code ICARUS est composé d'une multitude de scripts, dont la majorité manipule des tableaux de tableaux de tableaux de très grande taille. Les opérations étant relativement simples (étant surtout des additions/multiplications des différents termes de ces tableaux), et étant effectuées de manière linéaire, ma première tâche fut de regarder à quels endroits une parallélisation du code pouvait être envisageable dans ces différents fichiers, et comment cette parallélisation devait prendre place.

Sébastien Clesse ayant déjà un peu d'expérience avec openMP (une interface de programmation pour le calcul parallèle), et les clusters que nous avions à disposition proposant un support pour cette approche, nous avons décidé d'employer cette méthode afin de paralléliser les boucles que nous pouvions optimiser dans les différents scripts. L'emploi de cette méthode n'a présenté aucun problème, mais nous souhaitions voir jusqu'où nous pouvions pousser l'optimisation du code. A cette fin, nous avons décidé de regarder dans quelle mesure une parallélisation sur GPU était plus favorable.

6.2.4 Optimisation sur GPU

Dans le cadre d'une parallélisation sur GPU, nous avons décidé d'employer openACC (standard conçu pour permettre du calcul parallèle sur GPU), Sébastien Clesse ayant eu l'opportunité d'effectuer une formation dans son usage il y a quelques années. Personne au CURL ne semblant avoir déjà travaillé avec cet outil, nous avons dû procéder à l'installation sur le cluster des différents modules nécessaires au bon fonctionnement d'openACC.

La raison pour laquelle nous avions des doutes sur une meilleure efficacité des GPU sur les CPU dans notre cas est le fait que nous travaillions principalement avec des matrices. Or, les GPU basent principalement leur fonctionnement sur les calculs matriciels, leur structure étant elle-même pensée pour cela.

J'ai dès lors entamé des batteries de tests sur des opérations matricielles simples, dans un premier temps afin d'établir dans quelle mesure l'usage d'openACC devait être effectué de manière optimale. Par exemple, dans le cas de dix multiplications de matrice 1000x1000, il fallait compter environ 25 secondes afin d'effectuer ces opérations sans parallélisation. En revanche, à l'aide d'openACC, nous sommes parvenus à faire varier le temps d'exécution sur cluster entre 0.13 et 0.5 secondes, ce qui témoigne d'un gain conséquent, là où openMP nous permettait de réduire le temps d'exécution à environ 2.5 secondes.

L'usage d'openACC proposait deux méthodes différentes applicables à notre cas : l'utilisation d'une structure en noyaux (kernels) ou une parallèle. La structure en noyaux permettait basiquement de sélectionner une portion de code, et openACC produisait dès lors un nombre arbitraire de noyaux associé à un nombre arbitraire de dimensions qui étaient exécutés de manière séquentielle de manière à paralléliser la portion du code. La structure parallèle, elle, permettait une gestion plus fine du contrôle sur la façon avec laquelle le compilateur structure la charge de travail sur le GPU, en spécifiant par exemple les dimensions spécifiques pour la parallélisation.

Ainsi, l'emploi d'une méthode en noyaux octroie une liberté plus grande au compilateur, afin de trouver et faire correspondre un modèle de parallélisation à la portion de code souhaitée, là où une méthode "parallèle" restreindra beaucoup plus le compilateur, et nécessitera à celui qui l'implémente une étude plus approfondie du code à optimiser.

Dans notre situation, nous n'avons pas eu le temps d'explorer totalement l'usage de la méthode "parallèle". L'utilisation d'openACC à l'aide de la structure en noyaux nous donnait alors les meilleurs résultats, nous permettant donc dans l'exemple cité plus haut d'obtenir un temps d'exécution sur cluster de 0.13 secondes au lieu de 25 secondes.

6.3 Développement d'un programme de visualisation de cordes cosmiques

6.3.1 Introduction

La partie concernant la création du programme de visualisation de cordes cosmiques fut séparée en plusieurs étapes. Dans un premier temps, il m'a fallu explorer les possibilités et choisir le bon outil afin d'avoir une représentation des données adaptées. Ensuite, une fois une image générée à partir des données, l'idée était de faire du programme une animation, afin de pouvoir voir la simulation des différentes cordes cosmiques évoluer au cours du temps. Finalement, une dernière partie fut axée autour de l'interactivité présente entre la simulation et l'utilisateur. Un désir de Christophe Ringeval était de pouvoir se "balader" dans la simulation, afin de pouvoir avoir différents angles de vues. J'ai donc poli le programme, y ai rajouté divers paramètres, des contrôles, des raccourcis et une possibilité de choisir une coloration (établie au choix par l'utilisateur) afin de mettre en évidence les différentes caractéristiques des cordes cosmiques.

Cette partie fut accompagnée de ses contraintes techniques, me forçant à constamment retravailler et affiner le programme afin d'avoir un résultat satisfaisant. Ces contraintes étaient principalement liées à la taille des données. Effectivement, la simulation retraçait l'évolution des plusieurs milliers de cordes cosmiques, certaines de ces cordes possédant elles-mêmes plusieurs millions de points à représenter. La masse de données était donc colossale, et il fallait choisir le bon outil afin de garantir une certaine fluidité et un espace de stockage acceptable.

Christophe Ringeval m'a donné carte blanche vis-à-vis de la réalisation de ce projet, tant que les objectifs initiaux étaient atteints (à savoir avoir une simple représentation 3D des données acquises, dans un premier temps). Dès lors, j'ai opté pour l'utilisation du langage Python, qui s'avère être un langage qui propose une pléthore de possibilités vis-à-vis du traitement de données.

6.3.2 Lecture des données

La toute première étape obligatoire de ce projet, avant de pouvoir effectuer la moindre visualisation, était de lire les données brutes communiquées par Christophe Ringeval. Ces données étaient accessibles par le biais de fichiers stockés en format binaire, où chaque fichier correspondait à l'état de toutes les cordes cosmiques existantes à un instant t de la simulation. Certaines cordes cosmiques étant composées de plusieurs millions de points, la taille de ces fichiers avoisinaient généralement les 50Mo.

Christophe m'avait communiqué la structure des données. Pratiquement, chaque ligne de ce fichier .bin reprenait les caractéristiques d'un point composant une corde. On avait donc sa position dans des coordonnées cartésiennes, x , y et z étant toutes trois stockées dans des variables de type float de 4 bytes (type de variable usuellement nommé *f4*). On a également la longueur et l'âge de la corde contenant le point, stockés dans des variables de type *f4*. Finalement, on a l'identifiant de la corde associée au point, stocké dans un integer de 4 bytes (type appelé *i4*).

Mon rôle dans un premier temps a donc été d'effectuer une analyse syntaxique (parser, en anglais) des fichiers binaires afin de pouvoir stocker les valeurs des différents points dans des variables utilisables par le programme. Pour se faire nous pouvons soit lire manuellement le fichier byte par byte, afin d'en extraire les différentes variables au fur et à mesure, ou bien utiliser une bibliothèque afin d'effectuer cette première tâche. Par souci de simplicité, j'ai choisi d'utiliser la bibliothèque "numpy", bibliothèque extrêmement répandue dans la lecture et le traitement de données en Python. Numpy permet par exemple de lire un fichier en spécifiant la structure des données, et de le stocker sous forme d'un tableau (structure de donnée "ndarray" introduite par la librairie).

Il m'a été demandé, si possible, d'effectuer une première analyse des données, afin de vérifier que ces dernières soient cohérentes avec les résultats escomptés (pas de corde composée d'un seul point, âge de la corde identique pour chacun des points de cette corde, positions des données restreintes au domaine prévu, etc...). Cette analyse a pu révéler que chacun des fichiers comprenait bizarrement à chaque fois un point esseulé, associé à une corde qui n'était composée que de ce seul point. Ce problème a vite été réglé du côté de Christophe, nous garantissant ainsi que la structure des données était désormais bien cohérente avec les résultats souhaités.

Une contrainte qui est vite apparue plus tard dans le projet, vis-à-vis de la lecture des données, était le fait que la masse de données à lire était colossale, la masse totale des données étant de plus d'un To. Il était dès lors impensable de stocker tout en mémoire vive dans des variables, dans l'optique de lire ces variables lors de l'animation. Cela implique que les données devaient être lues au fur et à mesure que l'animation se déroulait. La simulation possédant un intérêt esthétique tout autant que scientifique (celle-ci ayant entre autre pour but de pouvoir être utilisée lors de conférences afin de donner un aperçu des résultats obtenus), il était capital d'assurer une certaine qualité dans la fluidité de l'animation. Dès lors, je me suis fixé un objectif correspondant à une fluidité de 30fps (c'est-à-dire afficher 30 images par seconde, ou bien autrement dit lire 30 fichiers binaires par seconde). Chaque fichier faisant approximativement 50Mo, et les fichiers devant être lus au fur et à mesure, cela correspondait à une masse de données lues équivalente à 1.5Go par seconde, ce qui est colossal quand on sait que ces dernières données devront être traitées de manière à être affichables dans le même intervalle de temps.

Cela nous emmène à la contrainte majeure à laquelle j'ai du faire face lors de la réalisation du programme : l'optimisation de la fluidité de la simulation. Cette optimisation ne pouvait malheureusement pas être réglée aisément en ne travaillant que sur une seule étape du workflow du programme : effectivement, chaque étape, de la lecture jusqu'à l'animation des données, devait être finement optimisée afin de pouvoir garantir la performance recherchée. Dans l'étape associée à la lecture des données, une des manières les plus évidentes d'aborder le problème était soit de réduire la masse de données à lire par seconde, soit d'aborder une structure de données plus rapide à lire par l'ordinateur. Cela nous emmène à la section suivante, portant sur un pré-traitement des données.

6.3.3 Pré-traitement des données

Dans le cadre d'une lecture de données plus aisée, j'ai dans un premier temps décidé de transformer les fichiers de données binaires d'origine en des fichiers beaucoup plus légers, en réduisant pour commencer la masse de données que ces dernières contenaient. Ainsi, le programme sera constitué de deux scripts : l'un en charge de transformer les données d'origine en des données plus faciles à lire, afin d'effectuer un premier traitement des données et de faciliter l'animation, dont le deuxième script sera en charge.

Dans le cadre de l'objectif principal du projet, seule la position des points était nécessaire, l'idée du projet étant d'afficher des nuages de millions de points, composant les cordes cosmiques. Le fichier de données transformé a donc été généré de manière à ne plus contenir que les coordonnées (x, y, z) de chaque point, les autres caractéristiques n'étant pas essentielles à leur représentation.

Cependant, lors de la réalisation d'un objectif secondaire (qui consistait à colorier les cordes en fonction de leur âge), l'usage de l'âge des cordes devenait nécessaire. Les problèmes qui sont apparus alors étaient issus de l'ajout de cette caractéristique dans le fichier de données

allégé qui augmentait considérablement la taille des données à lire par seconde, et le traitement pour déterminer la couleur des différents points prenait également un certain temps lors de l'animation. Aussi ai-je donc pris la décision de déterminer la couleur du point lors du pré-traitement, afin de ne plus avoir à le faire durant la simulation, quitte à augmenter davantage la taille des données à lire. Effectivement, après plusieurs expérimentations, je me suis aperçu que la lecture d'une valeur RGB associée à la couleur prenait moins de temps que la lecture d'un simple float codé en 4 bytes (pour l'âge) ajouté au temps nécessaire pour déterminer la couleur du point. Pour chaque fichier contenant les positions des différents points, j'ai donc créé un fichier contenant la couleur des différents points en RGB, ces seconds fichiers étant d'une taille d'environ 5Mo chacun.

La raison pour laquelle je n'ai pas fusionné les deux fichiers était que je souhaitais garantir la possibilité à l'utilisateur d'afficher ou non les couleurs. Dès lors, un utilisateur qui n'affiche pas les couleurs n'aurait besoin que du fichier correspondant aux positions des points, abaissant ainsi la quantité de données lues, et augmentant par conséquent la performance de la simulation. Cela peut entre autre être motivé par le fait que la simulation soit tournée sur des vieux ordinateurs, ne possédant pas spécialement des composants très puissants. J'ai préféré donc pouvoir laisser la possibilité à l'utilisateur de choisir les différents paramètres (comme nous le verrons plus loin) régissant la simulation afin de pouvoir abaisser la charge de travail de l'ordinateur.

Nous avons abordé le problème sous le spectre de la taille des données, mais regardons à présent la manière dont nous pouvons rendre les données plus faciles à lire. Cette partie fut plus complexe à aborder, et ne fut résolue qu'après avoir exploré les possibilités accordées par le moteur 3D que j'ai finalement choisi d'adopter : `openGL`. Il s'agissait ici de transformer le type de données pour les positions (3 float codés sur 4 bytes) en une structure plus aisément lisible pour l'ordinateur.

J'ai choisi d'employer une structure de données appelée "Vertex Buffer Objects", ou "VBOs". Les VBOs sont en `OpenGL` une méthode qui permet d'envoyer des données 3D vers la carte graphique. Cette méthode permet de faire en sorte que les calculs nécessaires à l'affichage d'images en 3D soient effectués non pas par le processeur, mais par la carte graphique qui est équipée d'unités de calculs spécialisées à cette fin. Tout comme pour les positions, les valeurs RGB des couleurs purent également être transformées en VBOs. Cela ne réduit nullement la taille des données en question, mais le fait que cette transformation de données soit effectuée au préalable dans le script de pré-traitement permet d'éviter d'avoir à le faire durant la visualisation. Ainsi, nous stockerons directement les variables sous format VBO à l'aide de la librairie "pickle", nos fichiers étant dès lors composés d'un fichier contenant les positions de tous les points à un instant t sous format VBO, et un fichier contenant les couleurs RGB de ces points, également sous format VBO. Notre script de visualisation n'aura dès lors plus qu'à lire les données, qui conserveront le format VBO à l'aide de la librairie pickle (c'est-à-dire que l'on n'aura plus à parser le fichier à l'aide de `numpy`, pickle permettant directement de stocker des variables Python dans des fichiers, nous faisant à nouveau gagner un peu de temps), et de les envoyer vers la carte graphique pour procéder à leur visualisation.

Voilà qui clôture les différents raisonnements qui m'ont permis de créer le premier script de pré-traitement, qui en conclusion lira les données d'origine, et produira pour chaque fichier deux nouveaux fichiers : un essentiel contenant les positions des différents points sous un nouveau format, et un autre contenant la couleur de ces différents points. Ces fichiers ont pour but de représenter une masse de données plus légère et plus facile à lire que celle d'origine.

6.3.4 Compression des données

Une éventualité qui été mentionnée avec Christophe est le fait que la masse totale des données, même allégée, correspond tout de même à une masse considérable (600Go), ce qui rend ces données très difficiles à transporter. Pour cela, j'ai déjà effectué une première recherche sur une potentielle méthode pour réduire la masse totale des données. Christophe proposait de regarder des façons de compresser les données afin de garantir une certaine praticabilité.

Plusieurs algorithmes s'offrent alors à nous ; mais, après une brève recherche, je me suis vite aperçu que la vitesse de décompression de ces algorithmes les rendait complètement inutilisables durant la simulation (les algorithmes principaux étudiés étant ici Gzip, Bzip2 et XZ). Sur un bon ordinateur, ces algorithmes ne dépassent que rarement une vitesse de décompression de 100Mb/s, ce qui est très loin d'être suffisant pour notre cas.

Cependant, ces algorithmes demeurent utiles afin de stocker les données, sans avoir comme objectif de les lire dans l'immédiat (afin de minimiser leur espace pris sur les serveurs, par exemple). Pour cela, utiliser la méthode de compression "XZ" semble être celle qui propose un taux de compression le plus optimal, tout en assurant qu'aucune perte n'est effectuée durant la compression. De plus, notre algorithme peut être couplé à un algorithme ré-arrangeant nos données afin de garantir un taux de compression encore plus satisfaisant. Par exemple, l'algorithme de Burrows-wheeler est une méthode qui permet de ré-arranger nos données de manière à augmenter les probabilités que des répétitions successives prennent place dans ces dernières. Cela permettrait donc d'obtenir une taille de fichier encore plus faible, tout en conservant les mêmes données.

6.3.5 Visualisation

Dans un premier temps, ayant un peu d'expérience dans la modélisation 3D à l'aide du moteur graphique Unity, et n'étant pas encore sûr du langage de programmation que j'allais aborder, c'est dans cette direction que je me suis d'abord lancé. Cependant, je me suis vite rendu compte que ce moteur était bien plus adapté à la représentation d'objets définis par des "surfaces" que pour la représentation de nuages massifs de points. Je me suis alors tourné vers les différentes bibliothèques accessibles dans le langage Python.

Une première approche du langage Python, pour le rendu d'une image fixe, fut effectuée à

l'aide de la librairie matplotlib, mettant à disposition la possibilité de rendre en 3D un nuage de points. Mes premiers tests n'étaient pas centrés sur la réalisation d'une animation, mais simplement sur l'affichage des données contenues dans un seul fichier, c'est-à-dire du nuage de points correspondant aux cordes cosmiques à un instant t . Cependant, la charge de données que représentait l'ensemble des points rendait très long la génération de ces nuages de points, même si le rendu était acceptable. Ce temps rendait complètement in-envisageable le fait d'en faire une animation à l'aide des différents outils mis à disposition par matplotlib.

Je me suis donc alors intéressé à OpenGL : une librairie multi-langages orientée dans le rendu 2D/3D. J'ai trouvé sans grande difficulté un support Python non-officiel de cette librairie, et réussi à avoir un premier résultat satisfaisant, présentant une image 3D fixe correspondant aux cordes cosmiques visibles à un instant t .

Je me suis alors attelé à la réalisation d'une animation, notre objectif étant alors un rendu correspondant à 30 images par seconde (30 fps). Un tel rendu demandait de lire, charger en mémoire, et afficher 1.5 Go de données par seconde, chaque fichier de cordes cosmiques correspondant à un instant t à la taille d'environ 50 Mo. Bien que l'animation fut possible à l'aide d'OpenGL, cette charge de données ruina les performances de la visualisation, les faisant baisser à environ 5 fps. C'est alors que la réalisation d'un script de pré-traitement, introduit plus haut, fut nécessaire, nous permettant d'avoir accès à une performance plus acceptable. Ce script de pré-traitement, couplé avec l'usage des VBOs, fit passer la charge de données à visualiser par seconde de 1.5Go à 600Mo, garantissant une fluidité dépassant les 30 fps en fonction de la machine utilisée.

En le script de visualisation consistait alors à aller chercher, 30 fois par seconde, un fichier correspondant à la position des différents points à afficher en format VBO. Une fois un fichier lu, le VBO était directement envoyé à la carte graphique, qui pouvait directement l'interpréter afin d'afficher le nuage de points en question.

6.3.6 Interface utilisateur

L'animation et le script de pré-traitement étant terminés, je me suis attaqué aux différentes interactions possibles entre l'utilisateur et la simulation, qui relevaient des objectifs secondaires proposés par Christophe Ringeval.

Pour ces différentes fonctionnalités, j'ai pris la décision de permettre à l'utilisateur de les régler à l'aide du clavier, ou à l'aide de la souris (via un panneau dans une fenêtre à part dédiée aux paramètres de la simulation). Cela fut rendu possible en incorporant l'affichage OpenGL dans une fenêtre disponible à l'aide de la bibliothèque pygame. Cette fenêtre a l'avantage de proposer une possibilité d'incorporer des éléments d'OpenGL (dédié à l'affichage 3D) et des éléments tkinter (dédié à l'affichage de menus, boutons, etc...). La fenêtre principale du programme était alors composée de deux parties : une affichant le résultat graphique des cordes cosmiques, à l'aide d'OpenGL, et une permettant de régler à la souris la totalité des paramètres régissant la simulation, chacun de ces paramètres étant également réglable au

clavier.

Un premier des objectifs secondaires était de proposer un système de contrôle de la caméra, permettant à l'utilisateur de se déplacer dans le nuage de points, comme si il y était lui-même, à l'aide de différentes touches du clavier. Heureusement, OpenGL met à disposition des outils permettant d'y parvenir assez facilement. Un problème que j'ai rapidement rencontré est le problème de la sensibilité des déplacements, étant donné que l'échelle des cordes cosmiques est très contrastée, d'une corde à l'autre. Il était du coup parfois très difficile de zoomer sur des petites cordes, c'est pourquoi j'ai implémenté une fonctionnalité qui permettait de régler manuellement la sensibilité de la caméra.

Ces contrôles n'influent en rien sur la fluidité de la visualisation, et permet à l'utilisateur d'adopter aisément le point de vue souhaité, de zoomer sur les différentes cordes, le tout pendant que l'animation se déroule.

Cela effectué, j'ai ensuite travaillé sur la possibilité d'effectuer des pauses dans la simulation, afin de pouvoir mettre en évidence certains événements au cours d'une présentation. Cependant, il y avait certains phénomènes qui étaient extrêmement rapides, ce qui rendait très difficile de s'arrêter, ou simplement d'observer ces événements. J'ai alors pris l'initiative d'ajouter une jauge qui permet manuellement de régler la vitesse à laquelle la simulation s'écoule, permettant ainsi d'avoir par exemple une simulation qui tourne au ralenti ou qui est très accélérée en fonction des éléments que l'on veut mettre en évidence. Pour compléter cet aspect de la simulation, j'ai également octroyé à l'utilisateur la possibilité de sélectionner lui-même l'instant t qu'il souhaite observer, en rentrant manuellement cet instant dans le menu créé à cet effet.

Ces différentes fonctionnalités permettent également à l'utilisateur d'épargner un peu la puissance requise par son ordinateur pour faire tourner la simulation, si son ordinateur n'est pas très performant. En effet, la simulation est pensée de manière à ce que si elle est mise au ralenti, le nombre de fichiers lus par seconde sera diminué en conséquence. Cependant, si la simulation est trop accélérée, j'ai pris la précaution de limiter le nombre de fichiers lus par seconde, afin de ne pas risquer de pertes conséquentes de performance à ce niveau-là.

Finalement, pour clôturer le projet, j'ai implémenté une fonction permettant de colorier les différentes cordes cosmiques en fonction de leurs paramètres, créant ainsi un fichier de couleurs par fichier de données de cordes lors du script de pré-traitement. Bien que la lecture de ce fichier impacte un petit peu sur la simulation (occasionnant généralement une perte de 5 fps), il est toujours possible pour l'utilisateur d'activer ou désactiver l'affichage des couleurs, afin de préserver les performances. Par défaut, la fonction définissant les couleurs colore les cordes en fonction de leur longueur et de leur âge, en mettant en évidence les cordes venant de faire leur apparition, c'est-à-dire les plus jeunes.

6.3.7 Manuel d'utilisateur

Accompagnant le programme, j'ai procédé à la création d'un manuel de l'utilisateur, expliquant en détails l'installation et l'utilisation des différents scripts à faire tourner, ainsi que contenant les détails concernant la licence et les pré-requis du programme. J'ai également produit une version parallélisée du script de pré-traitement, permettant ainsi de l'exécuter sur cluster pour traiter de grandes masses de données. Le manuel figure ci-dessous, clôturant ainsi ce chapitre abordant mon parcours effectué lors de mon séjour au cyclotron de Louvain-La-Neuve.

Manuel d'utilisateur

CSV : Cosmic Strings Visualisation

Ringeval Christophe et Welcomme Olivier
2019

Introduction

Cosmic String Visualisation est un script python dont le principal but est d'offrir une visualisation de nuages de points. Le cas le plus approprié au programme consiste ici en la représentation de points formant une corde cosmique, les coordonnées et les diverses caractéristiques de chacun de ces points étant générés dans un programme annexe.

La visualisation à proprement parlée prendra place en une animation, présentant l'évolution des différentes cordes cosmiques à travers le temps. Différentes fonctionnalités seront dès lors accessibles à l'utilisateur, que ce soit une gestion complète de la prise de vue, la création d'une potentielle charte de couleurs affichable au cours de la visualisation, un contrôle manuel de la vitesse à laquelle le temps s'écoule, et du choix de l'instant t de la visualisation que l'on souhaite observer.

Ce manuel ne comprend que les indications concernant la simple utilisation du programme. Si un besoin d'effectuer des changements, employer une autre structure de données ou ajouter divers choses se présente, une description détaillée du code est disponible dans le manuel du développeur, ainsi que dans les commentaires du dit code.

Conditions d'utilisation

La création du programme a été effectuée sur un environnement python 3.7.1, issu d'un package anaconda (Anaconda étant simplement un package contenant une version de python à laquelle sont déjà ajoutés une grande quantité de librairies scientifiques), une librairie faisant office d'adaptation python d'opengl ayant été utilisée en plus des librairies innérentes à anaconda.

- **license**
 - Copyright (c) 2019, Christophe Ringeval (and Olivier Welcomme)
 - Licence: GPLv3 or higher <http://www.gnu.org/licenses/gpl.html>
- **Prérequis**
 - Une version python 3.* ,
 - La librairie Pygame, hébergeant le widget opengl,
 - La librairie PyOpenGL, pour l'aspect graphique des données,
 - La librairie tkinter, offrant une fenêtre pour les différents paramètres,
 - La librairie numpy, pour une structure de données sous forme de matrice,
 - La librairie Pickle, pour les différents transferts de fichiers,
 - La librairie PyQt5, liant Pygame et PyOpenGL,
 - Les différentes librairies normalement installées de base : math, os et time.

Comme dit plus haut, une installation d'Anaconda et d'OpenGL pour python devrait suffir à l'exécution du programme, mais en cas d'erreur, veuillez vous référer à la liste ci-dessus afin de voir si toutes les librairies sont bien présentes.

Installation

Outre l'installation au préalable des différentes librairies, le programme ne nécessite aucune autre installation.

Le programme se divise en deux scripts python :

- CSV.py : ce script est la partie principale du programme, son utilisation réside en la visualisation à proprement parlée des différentes cordes cosmiques. L'obtention des données traitées devant être effectuée au paravant à l'aide du script traitement.py. Ces données traitées devront être situées dans le dossier "data-strings", lui-même situé dans le dossier racine. Ces données devront être nommées "[nom].bin_data" pour les coordonnées des points, et "[nom].bin_color" pour la charte de couleur associée à chaque point (le script traitement.py les nommant ainsi par défaut).
- traitement.py : ce script, quant à lui, est en charge de lire les données brutes des cordes cosmiques, et d'effectuer un pré-traitement afin de garantir une certaine performance lors de la visualisation. Pour son exécution, il est capitale que les données sources des différentes cordes cosmiques soient situées dans le dossier "strings", lui-même situé dans le dossier racine. Le dossier "data-strings", aussi situé dans la racine, devra également être créé afin d'accueillir les données traitées. Une version parallélisée, pour les clusters, est également disponible, le manuel du développeur détaillera les modifications à effectuer afin d'avoir accès à son utilisation. Les données initiales devant être de la forme "[nom].bin", dans le dossier "strings". **Il est important de créer le dossier "data-strings" avant d'exécuter le script.**

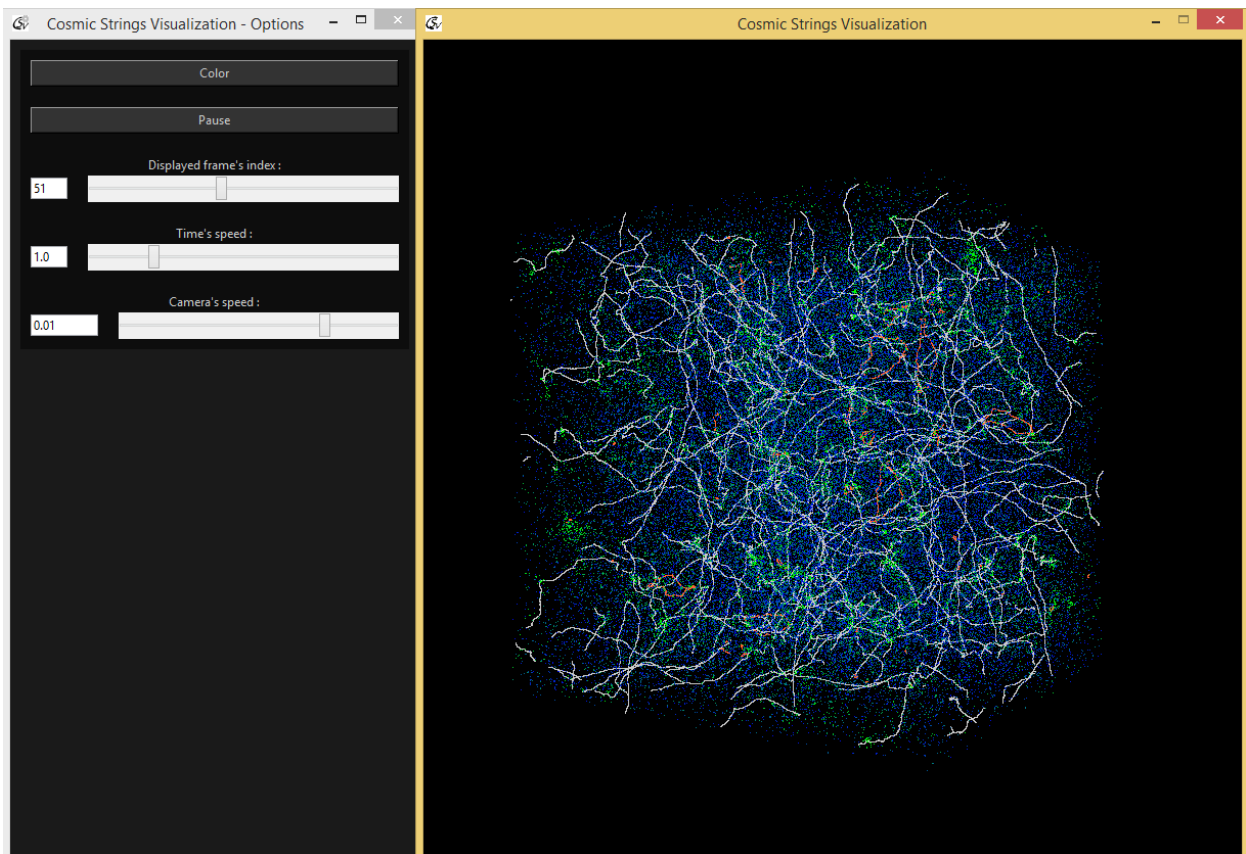
Utilisation

- traitement.py : s'agissant d'un script python, son utilisation s'effectue à l'aide de la commande "python traitement.py", en terminal. Comme cité plus haut, son utilisation requiert que les deux dossiers "strings" et "data-strings" soient créés au préalable dans le dossier racine, et que le dossier "strings" contiennent les données brutes nommées de la forme "[nom].bin", et dont la structure est la suivante :

```
[('x', '<f4'), ('y', '<f4'), ('z', '<f4'), ('length', '<f4'), ('time', '<f4'), ('id', '<i4')]
```

L'exécution peut durer un certain temps, voir plusieurs heures si la quantité de données à traiter est très volumineuse. Un pourcentage devrait être affiché, estimant l'avancée du programme. Afin d'assurer un gain de temps, une version parallélisée peut être obtenue et tournée sur cluster. Cette alternative est développée dans le manuel du développeur. Une fois l'exécution du programme achevée, l'utilisation du script CSV.py peut être effectuée.

- CSV.py : tout comme le script précédent, son utilisation est également effectuée à l'aide de la commande "python CSV.py" en terminal. Le programme devrait prendre quelques dizaines de secondes à se lancer avant de laisser apparaître deux fenêtres côte à côte :



- La fenêtre de gauche, dotée de deux boutons et trois sliders, correspond aux différents paramètres utilisable par l'utilisateur afin d'influer sur la visualisation en cours, visible sur la fenêtre de droite :
 - Le premier bouton, le bouton "color", permet d'utiliser la charte de couleur définie en fonction de l'exécution du script traitement.py. Si le bouton à un fond blanc, aucune charte de couleur n'est adoptée et toutes les cordes cosmiques seront de couleur identique. En cliquant dessus, le fond du bouton deviendra noir et les cordes cosmiques adopterons les couleurs définies au préalable. L'utilisateur peut à tout moment re-cliquer sur le bouton afin de revenir à l'état initial. Le raccourcis "c" permet également d'activer le bouton, si le focus est sur la fenêtre de droite, c'est-à-dire si le dernier clic a été effectuée sur cette dernière. Il est important de noter qu'activer les couleurs provoquera une légère perte de performance, le nombre de données à lire étant un peu plus élevé.
 - Le second bouton aura pour seule utilité de mettre la simulation en pause, permettant ainsi de rester sur la "frame" actuelle. La pause n'empêchera en rien l'utilisation des autres paramètres, ou la modification de la caméra. A l'instar du premier bouton, le fond du bouton sera noir lorsque la pause sera active, ou blanc lorsque la simulation s'écoulera de

manière constante. A nouveau, si le focus est sur la visualisation (fenêtre de droite), le raccourcis "touche espace" permettra d'activer et de désactiver la pause. Activer la pause impliquera qu'aucune autre donnée n'a besoin d'être lue, et par conséquent une très grande hausse de performance et de fluidité. Ainsi, il est conseillé d'ajuster la caméra lorsque la pause est active afin d'éviter trop de saccades dans les déplacements.

- Le premier slider indique l'indice de la frame affichée actuellement. Autrement dit, il affiche le quantième est le fichier de donnée en cours de lecture et de visualisation. L'utilisateur pourra bien entendu changer manuellement la frame affichée, en écrivant dans l'entrée prévue à cet effet et en appuyant sur la touche "enter" pour valider l'entrée, ou en faisant glisser la barre accompagnant le slider. Il est important de noter que de telles modifications ne peuvent être effectuées que lorsque la simulation est en pause, afin de n'engendrer aucun conflit. Ainsi, faire glisser manuellement le slider permettra aussi de parcourir visuellement la totalité de la simulation, si désiré.
 - Le second slider, lui, définira la vitesse à laquelle la simulation s'écoule, "1" étant choisi arbitrairement comme vitesse par défaut. Ainsi, une vitesse de 2 fera s'écouler la simulation deux fois plus vite, et une vitesse de 0 aura le même effet que de laisser la pause activée, à l'exception qu'une hausse de performance ne pourra pas spécialement être remarquée, dans ce cas là. Ce slider peut être modifié à tout moment, de la même manière que le premier slider, c'est-à-dire soit en faisant glisser la barre associée au slider, soit en entrant manuellement une valeur et en appuyant sur "enter" afin d'actualiser les données.
 - Le dernier slider sera quant à lui associé à la vitesse de déplacement de la caméra. Contrairement aux autres sliders, celui-ci ne sera pas associé à une échelle linéaire, mais à une échelle logarithmique base 10. Ainsi, il sera plus aisé d'effectuer des déplacements très précis si besoin, ou bien de naviguer rapidement à travers toute la simulation si souhaité. La valeur par défaut étant de 0.1, avoir une valeur de 0.01 permettra de se déplacer dix fois plus lentement, et une valeur de 1e-5 permettra de se déplacer 10.000 fois plus lentement. Ce slider est également modifiable à tout moment, que ce soit en faisant glisser ou en insérant la valeur souhaitée (toujours en appuyant sur enter pour actualiser par après). De plus, les raccourcis "+" et "-" du pavé numérique permettront, lorsque que le focus est une fois de plus pris sur la simulation (en cliquant sur la fenêtre de droite), de faire glisser le slider vers la gauche ou la droite, de manière à aller dix fois plus vite/lentement à chaque utilisation du raccourcis. Ces raccourcis sont extrêmement utiles lorsque l'on souhaite se rapprocher d'une corde en particulier, et zoomer dessus de plus en plus précisément.
- La fenêtre de droite sera associée à la visualisation à proprement parler. Prendre le focus dessus (cliquer dessus) permettra l'utilisation de nombreux raccourcis permettant tant de gérer la caméra de manière complète que d'avoir accès aux paramètres de la fenêtre de gauche. Voici les différents raccourcis accessibles à l'utilisateur lorsque le focus est obtenu :
 - **"espace"** : permet la mise en pause de la simulation.
 - **"z/q/s/d"** : déplace la caméra sur le plan horizontale, respectivement vers le haut/gauche/bas/droite.
 - **"r/f"** : déplace verticalement la caméra, respectivement vers le haut/bas, de manière à pouvoir gérer la totalité des déplacements à une main.
 - **"gauche/droite"** : Fais effectuer une rotation à la caméra, vers la gauche ou la droite, permettant de "tourner" le regard.
 - **"c"** : active ou désactive la charte couleur des cordes cosmiques.
 - **"+/-"** : permet d'augmenter ou de diminuer la vitesse des déplacements d'un facteur 10, permettant ainsi de gérer la caméra de manière précise.

Si certains de ces raccourcis ne sont pas adaptés au clavier ou à vos choix, il est aisément possible de ré-attribuer les touches de la manière souhaitée (se référer au manuel du développeur) ou de modifier la configuration du clavier en clavier azerty, si la configuration actuelle du clavier est en qwerty.

Bibliographie

- [1] URL : <http://tdc-www.harvard.edu/catalogs/bsc5.html>.
- [2] W. B. Bonnor. The photon rocket. *Class. Quant. Grav.* 11, 2007 (1994). URL : <https://doi.org/10.1088/0264-9381/11/8/008>.
- [3] T. Damour. Photon rockets and gravitational radiation. *Classical Quantum Gravity* 12, 725, 1995. URL : <https://doi.org/10.1088/0264-9381/12/3/011>.
- [4] A. Füzfa. Interstellar travels aboard radiation-powered rockets. *Phys. Rev. D* 99, 104081, 31 May 2019. URL : <https://doi.org/10.1103/PhysRevD.99.104081>.
- [5] E.ourgoulhon. *Relativité restreinte des particules à l'astrophysique*. EDP Sciences and CNRS Editions, Paris, France, 2010.
- [6] C. R. McInnes J. F. L. Simmons. was marx right? or how efficient are laser driven interstellar spacecraft? *American Journal of Physics*, Vol. 61, No. 3, March 1993.
- [7] W. Kinnersley. Field of an arbitrarily accelerating point mass. *Phys. Rev.* 186, 1335, 25 October 1969.
- [8] C. Lagoute and E. Davoust. The interstellar traveler. *American Journal of Physics*, Vol 63, Issue 3, 1995. URL : <https://doi.org/10.1119/1.17958>.
- [9] G. Marx. Interstellar vehicle propelled by terrestrial laser beam. *Nature*, July 2, 1966.
- [10] P. Lubin N. Kulkarni and Q. Zhang. Relativistic spacecraft propelled by directed energy. *The Astronomical Journal*, Volume 155, Number 4, 2018 March 19. URL : <https://iopscience.iop.org/article/10.3847/1538-3881/aaafd2>.
- [11] J. Podolský. Photon rockets in the (anti-)de sitter universe. *Phys. Rev. D* 78, 044029, 13 August 2008. URL : <https://doi.org/10.1103/PhysRevD.78.044029>.
- [12] R. A. Schwartz R. W. Lindquist and C. W. Misner. Vaidya's radiating schwarzschild. *Metric Phys. Rev.* 137, B1364, 1965. URL : <https://doi.org/10.1103/PhysRev.137.B1364>.
- [13] J. L . Redding. Interstellar vehicle propelled by terrestrial laser beam. *Nature*, February 11, 1967.
- [14] R.L.Forward. Roundtrip interstellar travel using laser-pushed lightsails. *J.Spacecraft vol 21 no 2*, 1984.
- [15] P. C. Vaidya. The gravitational field of a radiating star. *roc. Indian Acad. Sci. (Math. Sci.)* 33, 264, 1951. URL : <https://doi.org/10.1007/BF03173260>.